PCT/JP2004/004414

29. 3. 2004

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

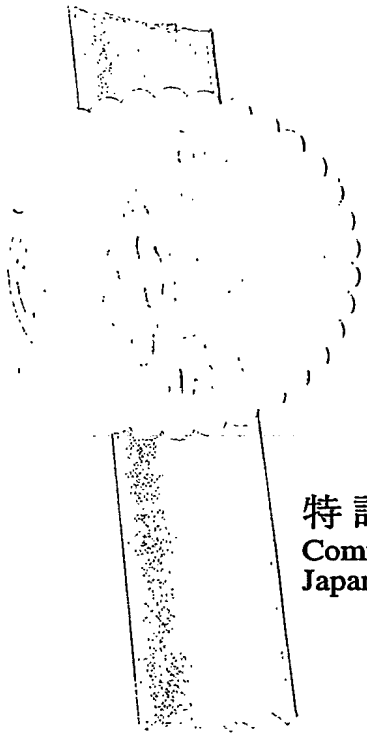
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２００３年 ４月１０日

出 願 番 号
Application Number: 特願２００３－１０６３９３

[ST. 10/C]: [ＪＰ２００３－１０６３９３]

出 願 人
Applicant(s): 松下電器産業株式会社



２００４年 ４月３０日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫

出証番号 出証特２００４－３０３７３０７

【書類名】　特許願
【整理番号】　2054041182
【提出日】　平成15年 4月10日
【あて先】　特許庁長官殿
【国際特許分類】　G06F　9/00　320
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　灘本　雄二
【特許出願人】
　【識別番号】　000005821
　【氏名又は名称】　松下電器産業株式会社
【代理人】
　【識別番号】　100097445
　【弁理士】
　【氏名又は名称】　岩橋　文雄
【選任した代理人】
　【識別番号】　100103355
　【弁理士】
　【氏名又は名称】　坂口　智康
【選任した代理人】
　【識別番号】　100109667
　【弁理士】
　【氏名又は名称】　内藤　浩樹
【手数料の表示】
　【予納台帳番号】　011305
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9809938

【書類名】　明細書

【発明の名称】　ウィンドウスタック制御方法

【特許請求の範囲】

【請求項１】　表示装置に複数のウィンドウを表示する際のウィンドウの重ね合わせを管理するウィンドウスタック制御方法であって、

１つ以上のウィンドウを前記表示装置に表示する１つ以上のアプリケーションプログラムと、

前記１つ以上のアプリケーションプログラムが表示するウィンドウの重ね合わせを管理するウィンドウ管理プログラムとを備え、

前記アプリケーションプログラムが前記ウィンドウ管理プログラムに対してウィンドウのグループを指定し、前記ウィンドウ管理プログラムが、前記アプリケーションプログラムからウィンドウの表示要求を受け、該ウィンドウの表示を行う際に、該ウィンドウのスタック順をグループ毎に一連とする制御を行うことを特徴とするウィンドウスタック制御方法。

【請求項２】　ウィンドウ管理プログラムは、グループ毎に纏めていない第1のウィンドウの表示要求をアプリケーションプログラムから受け取った際に、グループ毎に纏めていない全ウィンドウから、前記第1のウィンドウと同じグループに属する第1のウィンドウ群を、前記第1のウィンドウ群の中でのスタック順関係は保持したまま、既にグループ毎に纏められた前記第1のウィンドウと同じグループに属する第２のウィンドウ群とスタック順を一連にし表示処理を行うことを特徴とする請求項１に記載のウィンドウスタック制御方法。

【請求項３】　ウィンドウ管理プログラムは、グループ毎に代表ウィンドウを１枚生成し、ウィンドウのスタック順をグループ毎に一連とする際に、前記ウィンドウを前記代表ウィンドウの子ウィンドウとする事を特徴とする請求項１または２に記載のウィンドウスタック制御方法。

【請求項４】　ウィンドウ管理プログラムは、アプリケーションプログラムからの要求を受けて、グループ内でウィンドウスタックの最上位移動、最下位移動を行うことを特徴とする請求項１から３のいずれかに記載のウィンドウスタック制御方法。

【請求項５】　ウィンドウ管理プログラムは、アプリケーションプログラムからの要求を受けて、グループ単位でウィンドウスタックの最上位移動、最下位移動を行うことを特徴とする請求項１から３のいずれかに記載のウィンドウスタック制御方法。

【請求項６】　ウィンドウ管理プログラムは、第1のウィンドウ又は第1のグループに対するスタック変更要求をアプリケーションプログラムから受け取った際に、グループ毎に纏めていない全ウィンドウから、前記第1のウィンドウと同じグループ又は前記第1のグループに属する第1のウィンドウ群を、前記第1のウィンドウ群の中でのスタック順関係は保持したまま、既にグループ毎に纏められた前記第1のウィンドウと同じグループ又は前記第1のグループに属する第２のウィンドウ群とスタック順を一連にしてスタック変更を行うことを特徴とする請求項４または５に記載のウィンドウスタック制御方法。

【請求項７】　ウィンドウ管理プログラムは、アプリケーションプログラムによって指定されたウィンドウをグループに属させず、グループに属して表示された全てのウィンドウより常にスタック上位に位置させることを特徴とする請求項１に記載のウィンドウスタック制御方法。

【請求項８】　ウィンドウ管理プログラムは、最上位グループに属するウィンドウよりスタック下位のウィンドウを常に非表示状態とすることを特徴とする請求項１に記載のウィンドウスタック制御方法。

【請求項９】　ウィンドウ管理プログラムは、最上位グループに属するウィンドウの中のスタック最下位のウィンドウの直下に、特定のウィンドウを位置させることを特徴とする請求項１に記載のウィンドウスタック制御方法。

【請求項１０】　ウィンドウ管理プログラムは、Xウィンドウシステムとウィンドウマネージャとを備え、最上位グループの直上に特定のウィンドウを位置させることを特徴とする請求項４または５に記載のウィンドウスタック制御方法。

【請求項１１】　ウィンドウ管理プログラムは、Xウィンドウシステムとウィンドウマネージャとを備え、ウィンドウマネージャが、ウィンドウ破棄通知受信時に、ウィンドウシステム間とでスタック認識の整合性を確認し、異なっている場合は、ウィンドウシステムのスタック認識をウィンドウマネージャのスタック

認識にあわせ込む処理を行うことを特徴とする請求項１から３のいずれかに記載のウィンドウスタック制御方法。

【請求項１２】　ウィンドウ管理プログラムは、Xウィンドウシステムとウィンドウマネージャとを備え、ウィンドウマネージャが、ウィンドウシステムにスタック変更を要求する際にフラグを立てる一方で、ウィンドウ破棄通知受信時に、前記フラグが立っている時のみ、ウィンドウシステム間とでスタック認識の整合性を確認し、異なっている場合は、ウィンドウシステムのスタック認識をウィンドウマネージャのスタック認識にあわせ込む処理を行い、前記フラグを下げることを特徴とする請求項１から３のいずれかに記載のウィンドウスタック制御方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、コンピュータから表示装置に複数のウィンドウを表示する際のウィンドウの重ね合わせを管理するウィンドウスタック制御方法に関する。

【０００２】

【従来の技術】

近年、表示装置に複数のウィンドウを表示するコンピュータとして、ワークステーションや家庭用コンピュータが広く普及している。これらの場合、ユーザは、表示装置として比較的大きな画面に複数のウィンドウを表示して、マウスとキーボードを使ってウィンドウの重なりの上下関係であるスタック順の操作を行うのが一般的である。

【０００３】

例えば、Unix（R）ワークステーションで広く普及しているX11ウィンドウシステムの場合、ウィンドウの重なりの上下関係であるスタック順は、ウィンドウの生成時にスタック最上位に位置付けられ、マップされた後は、マウスやアプリケーションの操作によって、スタック位置を上下方向に移動させる。

【０００４】

具体例として、コンピュータ上で動作する３種類のアプリケーションが合計５

枚のウィンドウを表示装置であるスクリーンに表示する際の表示例となる模式図を図２８に示す。図２８において、１はCRTモニタや液晶モニタで実現される表示装置のスクリーン、２及び３は第１のアプリケーションが表示するウィンドウ、４及び５は第２のアプリケーションが表示するウィンドウ、６は第３のアプリケーションが表示するウィンドウであり、第１のアプリケーションが表示するウィンドウ２及びウィンドウ３を第１のグループ、第２のアプリケーションが表示するウィンドウ４及びウィンドウ５を第２のグループ、第３のアプリケーションが表示するウィンドウ６を第３のグループとする。

【０００５】

図２８を見ればわかるように、ウィンドウ３はウィンドウ２の上に重なり、ウィンドウ５はウィンドウ４の上に重なっている。この時のウィンドウスタック順の例となる模式図を図２９に示す。図２９を見ればわかるように、ウィンドウ３はウィンドウ２よりスタック上位にあるので、両者が表示上で重なる場合は、ウィンドウ３がウィンドウ２の上に重なる事となり、同様に、ウィンドウ５はウィンドウ４よりスタック上位にあるので、両者が表示上で重なる場合は、ウィンドウ５がウィンドウ４の上に重なって表示される。

【０００６】

そして、このような表示状態、スタック順状態で、第1のアプリケーションが何らかの都合で、例えば、ウィンドウ２をウィンドウ３の前面に表示させたくなった場合、ウィンドウ２をスタック最上位に持ってくる事で対応していた。この時のスタック順の模式図を図３０に示す。図３０を見ればわかるように、ウィンドウ２はウィンドウ３よりスタック上位にあるので、両者が表示上で重なる場合は、ウィンドウ２がウィンドウ３の上に重なって表示されることとなる。また、この操作により、ウィンドウ２が、第２のグループ、第３のグループのウィンドウを跨って、スタック順で下位方向から最上位へ移動した事がわかる。

【０００７】

【特許文献１】

特開昭６２－２９８８８２号公報

【０００８】

【発明が解決しようとする課題】

しかしながら、上記従来のウィンドウスタック制御方法においては、アプリケーションが或るウィンドウに対して、ウィンドウスタック移動、ウィンドウ生成及び表示を行った場合、グループを跨ってスタック制御をするため、意図せず、該ウィンドウが他のアプリケーションの表示するウィンドウの上に重なることがあるという問題があった。

【０００９】

上記課題に鑑み、本発明は、ウィンドウスタックの制御方法において、最前面でウィンドウを表示中のアプリケーションのウィンドウ表示に、他のアプリケーションが不用意に影響を及ぼす事の無いウィンドウスタック制御方法を提供することを目的とする。

【００１０】

【課題を解決するための手段】

この課題を解決するために本発明は、表示装置を含むコンピュータにおいて、前記表示装置に複数のウィンドウを表示する際のウィンドウの重ね合わせを管理するウィンドウスタック制御方法であって、アプリケーションプログラムがウィンドウにグループを指定し、ウィンドウ管理プログラムがウィンドウのスタック順をグループ毎に一連とする。

【００１１】

これにより、グループ最上位のウィンドウは、他のグループのウィンドウの下に重ねられて表示される事は無くなり、最前面でウィンドウを表示中のアプリケーションのウィンドウ表示に、他のアプリケーションが不用意に影響を及ぼす事を無くす事ができる。

【００１２】

【発明の実施の形態】

本発明の第１の発明は、表示装置に複数のウィンドウを表示する際のウィンドウの重ね合わせを管理するウィンドウスタック制御方法であって、１つ以上のウィンドウを前記表示装置に表示する１つ以上のアプリケーションプログラムと、前記１つ以上のアプリケーションプログラムが表示するウィンドウの重ね合わせ

を管理するウィンドウ管理プログラムとを備え、前記アプリケーションプログラムが前記ウィンドウ管理プログラムに対してウィンドウのグループを指定し、前記ウィンドウ管理プログラムが、前記アプリケーションプログラムからウィンドウの表示要求を受け、該ウィンドウの表示を行う際に、該ウィンドウのスタック順をグループ毎に一連とする制御を行うことを特徴とするウィンドウスタック制御方法であり、スタック順がグループ単位で纏められるため、グループ最上位のウィンドウは、他のグループのウィンドウの下に重ねられて表示される事が無い。

【００１３】

また、本発明の第２の発明は、第１の発明において、ウィンドウ管理プログラムは、グループ毎に纏めていない第1のウィンドウの表示要求をアプリケーションプログラムから受け取った際に、グループ毎に纏めていない全ウィンドウから、前記第1のウィンドウと同じグループに属する第1のウィンドウ群を、前記第1のウィンドウ群の中でのスタック順関係は保持したまま、既にグループ毎に纏められた前記第1のウィンドウと同じグループに属する第２のウィンドウ群とスタック順を一連にし表示処理を行うことを特徴とするウィンドウスタック制御方法であり、表示要求によって同一グループ内でみたスタック順が変更されることがないので、アプリケーションの同一グループ内のスタック順管理の負担が軽減される。

【００１４】

また、本発明の第３の発明は、第１の発明又は第２の発明において、ウィンドウ管理プログラムは、グループ毎に代表ウィンドウを１枚生成し、ウィンドウのスタック順をグループ毎に一連とする際に、前記ウィンドウを前記代表ウィンドウの子ウィンドウとする事を特徴とするウィンドウスタック制御方法であり、グループ単位のスタック移動やグループ単位のウィンドウの表示/非表示の切り替えを代表ウィンドウに対する制御で実施できる分、グループ内のウィンドウが多いほど、処理負荷を軽減することができるようになる。

【００１５】

また、本発明の第４の発明は、第1の発明又は第２の発明又は第３の発明にお

いて、ウィンドウ管理プログラムは、アプリケーションプログラムからの要求を受けて、グループ内でウィンドウスタックの最上位移動、最下位移動を行うことを特徴とするウィンドウスタック制御方法であり、スタックの操作がグループ内で閉じるので、各アプリケーションプログラムは、他のアプリケーションプログラムのウィンドウ表示を気にする事無くスタック変更を行う事ができる。

【００１６】

また、本発明の第５の発明は、第1の発明又は第２の発明又は第３の発明において、ウィンドウ管理プログラムは、アプリケーションプログラムからの要求を受けて、グループ単位でウィンドウスタックの最上位移動、最下位移動を行うことを特徴とするウィンドウスタック制御方法であり、ウィンドウの操作をグループ指定で実施できるので、例えば、グループをアプリケーションプログラム単位で振り分けた場合だと、そのアプリケーションに属するウィンドウのメンバーを知らなくても、スタック最前面にそのアプリケーションに属する全通常ウィンドウを移動させる事が可能となる。

【００１７】

また、本発明の第６の発明は、第４の発明又は第５の発明において、ウィンドウ管理プログラムは、第1のウィンドウ又は第1のグループに対するスタック変更要求をアプリケーションプログラムから受け取った際に、グループ毎に纏めていない全ウィンドウから、前記第1のウィンドウと同じグループ又は前記第1のグループに属する第1のウィンドウ群を、前記第1のウィンドウ群の中でのスタック順関係は保持したまま、既にグループ毎に纏められた前記第1のウィンドウと同じグループ又は前記第1のグループに属する第２のウィンドウ群とスタック順を一連にしてスタック変更を行うことを特徴とするウィンドウスタック制御方法であって、スタック変更対象となるグループに属する通常ウィンドウ全てを、その中でのスタック順は保持したままで纏めた上でスタック変更を実施するので、アプリケーションプログラムは、自グループのウィンドウのスタック順を把握し易くなる。

【００１８】

また、本発明の第７の発明は、第１の発明において、ウィンドウ管理プログラ

ムは、アプリケーションプログラムによって指定されたウィンドウをグループに属させず、グループに属して表示された全てのウィンドウより常にスタック上位に位置させることを特徴とするウィンドウスタック制御方法であって、グループ単位のスタック順に関係無くウィンドウを表示できるので、最優先で表示するウィンドウのサービスを提供することができる。

【００１９】

また、本発明の第８の発明は、第１の発明において、ウィンドウ管理プログラムは、最上位グループに属するウィンドウよりスタック下位のウィンドウを常に非表示状態とすることを特徴とするウィンドウスタック制御方法であって、最上位グループよりスタック下位のグループが何を表示しようとしても表示装置には表示されないので、最上位グループのウィンドウで覆われない部分に中途半端に映り込む心配が無い。

【００２０】

また、本発明の第９の発明は、第１の発明において、ウィンドウ管理プログラムは、最上位グループに属するウィンドウの中のスタック最下位のウィンドウの直下に、特定のウィンドウを位置させることを特徴とするウィンドウスタック制御方法であって、例えば、前記特定のウィンドウを表示装置のスクリーン一杯に表示しておけば、前記特定のウィンドウのスタック下位のグループが何を表示しようとしても，表示装置に表示されないので、最上位グループのウィンドウで覆われない部分に中途半端に映り込む心配が無い。

【００２１】

また、本発明の第１０の発明は、第４の発明又は第５の発明において、ウィンドウ管理プログラムは、Xウィンドウシステムとウィンドウマネージャとを備え、最上位グループの直上に特定のウィンドウを位置させることを特徴とするウィンドウスタック制御方法であって、最上位グループを別のグループに入れ替える際、Xウィンドウシステムの場合、XRestackWindows（R）関数を使用する事となるが、ウィンドウ群のスタック移動先を示すために使用できる。もし、最上位グループのウィンドウよりスタック上位にウィンドウが存在しなかった場合、存在する全てのウィンドウの並び順を決め、前記XRestackWindows（R）の引数に代

入せねばならないが、前記特定のウィンドウがあれば、前記特定のウィンドウと前記ウィンドウ群を引数に代入するだけで済む。

【００２２】

また、本発明の第１１の発明は、第１の発明又は第２の発明又は第３の発明において、ウィンドウ管理プログラムは、Xウィンドウシステムとウィンドウマネージャとを備え、ウィンドウマネージャが、ウィンドウ破棄通知受信時に、ウィンドウシステム間とでスタック認識の整合性を確認し、異なっている場合は、ウィンドウシステムのスタック認識をウィンドウマネージャのスタック認識にあわせ込む処理を行うことを特徴とするウィンドウスタック制御方法であって、ウィンドウ破棄通知受信時に整合性確保を図るので、ウィンドウシステムがスタック変更処理に失敗しても、処理が破綻する事がなくなる。

【００２３】

また、本発明の第１２の発明は、第１の発明又は第２の発明又は第３の発明において、ウィンドウ管理プログラムは、Xウィンドウシステムとウィンドウマネージャとを備え、ウィンドウマネージャが、ウィンドウシステムにスタック変更を要求する際にフラグを立てる一方で、ウィンドウ破棄通知受信時に、前記フラグが立っている時のみ、ウィンドウシステム間とでスタック認識の整合性を確認し、異なっている場合は、ウィンドウシステムのスタック認識をウィンドウマネージャのスタック認識にあわせ込む処理を行い、前記フラグを下げることを特徴とするウィンドウスタック制御方法であって、ウィンドウマネージャがスタック変更を実施した時のみ、スタック認識の整合性のための処理を行うので、処理の実行頻度を軽減できる。

【００２４】

以下、本発明の実施の形態について、図面を用いて説明する。

【００２５】

（実施の形態１）

本実施の形態では、ウィンドウ管理プログラムが、Xウィンドウシステムとウィンドウマネージャから成り、ウィンドウマネージャがルートウィンドウの子ウィンドウであるトップレベルウィンドウのスタック順を制御する場合を例にとっ

て説明する。

【００２６】

図１に本発明の実施の形態１のウィンドウスタック制御方法を実現するシステムの構成例を示す。

【００２７】

図１において、７はコンピュータ、８はCPU（Central Processing Unit）、９は各種プログラムや処理データ等を格納するメモリ、１０は各構成要素に所望の双方向通信を行わせるバス、１１はCRTモニタや液晶モニタといった表示装置、１は表示装置１１のスクリーン、１０１及び１０２，１０３はCPU８で実行され１枚以上のウィンドウをウィンドウ管理プログラム１０４を介してスクリーン１に表示しようとするアプリケーションプログラム、１０４はCPU８で実行されアプリケーションプログラムが表示しようとするウィンドウを管理してバス１０を介して表示を行うウィンドウ管理プログラムであり、ウィンドウシステム１２とウィンドウマネージャ１０５から成る、ウィンドウシステム１２は従来のXウィンドウシステム、１０５はウィンドウマネージャである。

【００２８】

以下、図面を用いて本実施の形態の動作を説明する。

【００２９】

まず、アプリケーションプログラム１０１、１０２、１０３は、ウィンドウ管理プログラム１０４を通じて、スタック制御特性の異なる２種類のトップレベルウィンドウを表示することができるものとし、その２種類を通常ウィンドウ、優先ウィンドウと呼ぶ事とする。両者の違いは、表示された優先ウィンドウは表示された全ての通常ウィンドウより常にスタック上位に表示されることと、通常ウィンドウはグループ単位でスタック順を一連とするが優先ウィンドウでは行わないことである。

【００３０】

アプリケーションプログラム１０１、１０２、１０３は、ウィンドウを表示する場合、ウィンドウシステム１２に対して、ウィンドウ生成要求、ウィンドウ種類及びグループ設定要求、マップ要求を発行する。この内、ウィンドウ種類及び

グループ設定要求については、Xウィンドウシステムの場合、プロパティ機構を用いる事で、ウィンドウ毎に値を設定することができる。

【００３１】

各ウィンドウは、ウィンドウ生成及びマップという過程を経て初めてスクリーン１に表示されるようになる。

【００３２】

ウィンドウシステム１２は、ウィンドウ生成要求を受け取ると、ウィンドウの重なりであるスタック順の最上位にそのウィンドウを位置させ、ウィンドウマネージャ１０５に対してウィンドウ生成通知イベント(CreateNotify)を発行する。

【００３３】

ウィンドウマネージャ１０５は、内部データベースとして、ウィンドウ毎にウィンドウ情報を保持する。

【００３４】

（表１）にウィンドウ情報の一例を示す。

【００３５】

【表１】

情報名	内容
ウィンドウ識別子	識別子
スタック位置確定フラグ	暫定/確定済み
スタック位置情報	位置情報
プロパティ獲得フラグ	未獲得/獲得済み
ウィンドウ種類	優先ウィンドウ/通常ウィンドウ
グループ値	グループの値

【００３６】

表１に示されるように、ウィンドウ情報は、ウィンドウ識別子、スタック位置確定フラグ、スタック位置情報、プロパティ獲得フラグ、ウィンドウ種類、グル

ープ値から成る。ウィンドウ識別子は、そのウィンドウ情報がどのウィンドウの情報かを示し、スタック位置確定フラグは、そのウィンドウが、ウィンドウマネージャにとって、スタック位置が暫定か確定済みなのかを示し、スタック位置情報は、そのウィンドウのスタック位置を示し、プロパティ獲得フラグは、そのウィンドウのウィンドウ種類とグループ値をウィンドウマネージャが獲得済みかどうかを示し、ウィンドウ種類は、前記プロパティ獲得フラグが獲得済みとなっている場合は、そのウィンドウが優先ウィンドウか通常ウィンドウのどちらであるかを示し、グループ値は、前記プロパティ獲得フラグが獲得済みとなっている場合は、そのウィンドウのグループ値を示す。

【００３７】

図１のウィンドウマネージャ１０５は、ウィンドウ生成通知イベント(CreateNotify)を受け取ることで、新規にウィンドウが生成された事を知り、そのウィンドウ生成通知イベント(CreateNotify)からウィンドウ識別子を獲得し、内部データベースに新規にウィンドウ情報格納領域を設け、そのウィンドウ情報のウィンドウ識別子に格納し、また、同ウィンドウのスタック位置確定フラグを暫定に、スタック位置情報をスタック最上位に、プロパティ獲得フラグを未獲得に、各々初期化する。

【００３８】

次に、ウィンドウシステム１２は、ウィンドウ種類及びグループ設定要求を受け取り、該ウィンドウのプロパティとして記憶する。

【００３９】

次に、ウィンドウシステム１２は、アプリケーションプログラム１０１や１０２、１０３からマップ要求を受け取り、内部処理は特にせずに、ウィンドウマネージャ１０５にマップ要求イベント(MapRequest)を発行する。

【００４０】

マップ要求イベント(MapRequest)を受け取った際のウィンドウマネージャ１０５の動作例となるフロー図を図２に示す。

【００４１】

図２に示されるように、S201では、該時点で受け取ったマップ要求イベント(M

apRequest)からマップ対象ウィンドウの識別子を獲得して、S202に進む。

【００４２】

S202では、マップ対象ウィンドウの識別子で内部データベースを検索し、対応するウィンドウ情報が存在する場合はS203へ進み、存在しない場合はS214へ進む。

【００４３】

S203では、内部データベースで対象ウィンドウのスタック位置確定フラグが確定済みかどうかをチェックし、確定済みの場合はS213に進み、暫定の場合はS204に進む。

【００４４】

S204では、対象ウィンドウのプロパティ獲得フラグをチェックし、獲得済みの場合はS206に進み、未獲得の場合はS205に進む。

【００４５】

S205では、対象ウィンドウのウィンドウ種類及びグループ値をウィンドウシステムに問い合わせて獲得し、内部データベースに格納し、S206に進む。

【００４６】

S206では、対象ウィンドウのウィンドウ種類をチェックし、優先ウィンドウの場合は、S213に進み、通常ウィンドウの場合は、S207に進む。

【００４７】

S207では、内部データベースに、対象ウィンドウと同じグループ値を持つスタック位置確定フラグが確定済みの通常ウィンドウがあるかどうかをチェックし、有る場合はS208進み、無い場合はS209に進む。

【００４８】

S208では、対象ウィンドウが、対象ウィンドウと同じグループ値を持つスタック位置確定フラグが確定済みの通常ウィンドウの内のスタック最上位のウィンドウの直上となるようウィンドウシステム１２にスタック変更要求を発行する。

【００４９】

S208のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図３に示す。

【００５０】

図３に示されるように、通常ウィンドウである対象ウィンドウが、対象ウィンドウと同じグループ値を持つスタック位置確定フラグが確定済みのウィンドウの内のスタック最上位のウィンドウの直上に移されているのが解る。 S209では、内部データベースから、通常ウィンドウ全体を対象にスタック位置確定フラグが確定済みのウィンドウを検索し、存在する場合はS210に進み、存在しない場合はS211に進む。

【００５１】

S210では、内部データベースにあるスタック位置確定フラグが確定済みの通常ウィンドウのうち、スタック位置が最上位のウィンドウの直上に対象ウィンドウが位置するよう、ウィンドウシステム１２に対してスタック変更要求を発行し、S212に進む。

【００５２】

S210のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図４に示す。

図４に示されるように、通常ウィンドウである対象ウィンドウが、スタック位置確定フラグが確定済みの通常ウィンドウ全体の内のスタック最上位のウィンドウの直上に移されているのが解る。

【００５３】

S211では、スタック位置の最下位に対象ウィンドウが位置するよう、ウィンドウシステム１２にスタック変更要求を発行し、S212に進む。

【００５４】

S211のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図５に示す。

図５に示されるように、通常ウィンドウである対象ウィンドウが、スタック最下位に移されているのが解る。

【００５５】

S212では、ウィンドウシステム１２に対してマップ対象ウィンドウのマップ要求を発行し、内部データベースの対象ウィンドウのウィンドウ情報の内、スタッ

ク位置確定フラグが暫定を示している場合は確定済みに更新し、また、内部データベース全体に対して必要に応じてスタック位置情報を更新した上で、S213に進む。

【００５６】

S213では、処理を終了する。

【００５７】

ウィンドウシステム１２は、図２のS208やS210、S211に起因するスタック変更要求をウィンドウマネージャ１０５から受け取り、要求に従ってウィンドウのスタック順を変更する。また、図２のS212に起因するマップ要求をウィンドウマネージャ１０５から受け取り、要求に従ってウィンドウのマップする。マップされたウィンドウは、ウィンドウの重なり具合にもよるが、他のウィンドウの下に隠れていなければ、スクリーン１に表示されるようになる。

【００５８】

図２の処理によって、アプリケーション１０１、１０２、１０３が生成及びウィンドウ種類及びグループ設定した通常ウィンドウは、表示までに必ず、スタック位置情報が確定済みとなり、また、スタック位置がグループ毎に纏められ、また、優先ウィンドウのスタック上位に表示される事もない。また、優先ウィンドウは、スタック位置が変更されることなく表示される。

【００５９】

図１において、アプリケーションプログラム１０１、１０２、１０３は、生成及びウィンドウ種類及びグループ設定した通常ウィンドウに対して、スタック変更要求として、グループ内スタック最上位移動をウィンドウシステム１２に対して発行する事ができる。Xウィンドウシステムの場合、例えば、グループ内スタック最上位移動要求にXRaiseWindow()関数を割り当てれば良い。また、優先ウィンドウに対して、スタック変更要求として、優先ウィンドウスタック最上位移動要求をウィンドウシステム１２に対して発行する事ができ、例えば、グループ内スタック最上位移動要求と同じXRaiseWindow()関数を割り当てる。

【００６０】

アプリケーションプログラムがXRaiseWindow()を実施した場合、ウィンドウシ

ステム１２はスタック変更をせず、ウィンドウマネージャ１０５に対してスタック最上位移動要求イベント（ConfigureRequest）を発行する。

【００６１】

スタック最上位移動要求イベントを受け取った際のウィンドウマネージャ１０５の動作例となるフロー図を図６に示す。

【００６２】

図６に示されるように、S601では、該時点で受け取ったスタック最上位移動要求イベント（ConfigureRequest）から対象ウィンドウの識別子を獲得して、S602に進む。

【００６３】

S602では、対象ウィンドウの識別子で内部データベースを検索し、対応するデータが存在する場合はS603へ進み、存在しない場合はS612へ進む。

【００６４】

S603では、内部データベースにあるプロパティ獲得フラグが未獲得である全ウィンドウについて、ウィンドウシステム１２に対してウィンドウ種類及びグループ値を問い合わせ、獲得できたウィンドウについては、そのウィンドウ種類及びグループ値とプロパティ獲得済みである事を内部データベースに記憶し、S604に進む。

【００６５】

S604では、対象ウィンドウのウィンドウ種類をチェックし、優先ウィンドウであればS605に進み、通常ウィンドウであればS606に進む。

【００６６】

S605では、内部データベースからウィンドウ種類が優先ウィンドウであるものの内、スタック最上位のウィンドウを検索し、そのウィンドウが対象ウィンドウ自身であれば、何もせずS612に進み、そのウィンドウが対象ウィンドウ自身でなければ、そのウィンドウの直上となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、内部データベース全体に対してスタック位置情報を更新した上で、S612に進む。

【００６７】

S605のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図７に示す。

【００６８】

図７に示されるように、対象ウィンドウのみが優先ウィンドウの中のスタック最上位のウィンドウの直上に移されているのが解る。

S606では、対象ウィンドウと同じグループに属する通常ウィンドウを内部データベースから検索してリストアップし、S607に進む。

【００６９】

S607では、前記リストアップした中にスタック位置確定フラグが確定済みであるウィンドウが存在するならばS608に進み、存在しないならS609に進む。

【００７０】

S608では、リストアップした中でスタック位置確定フラグが暫定であるウィンドウがあれば、それら全てを、それらの中でみたスタック順はそのままで、リストアップした中でスタック位置確定フラグが確定済みであるウィンドウの内の最上位のウィンドウの直上に位置させ、更に対象ウィンドウをリストアップしたウィンドウの中で最上位となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、リストアップした中でスタック位置確定フラグが暫定であるウィンドウがあれば確定済みに更新し、また、内部データベース全体に対してスタック位置情報を更新した上で、S610に進む。S608のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図８に示す。

【００７１】

図８に示されるように、対象ウィンドウと同じグループ値を持つスタック位置確定フラグが暫定のウィンドウが、同じグループ値を持つスタック位置確定フラグが確定済みのウィンドウの直上に移され、同一グループ値を持つウィンドウ全体の中で対象ウィンドウのスタック最上位移動がなされているのが解る。

【００７２】

図６において、S609では、内部データベースにスタック位置確定フラグが確定済みのウィンドウが存在するならばS610に進み、存在しないならばS611に進む。

【００７３】

S610では、S606でリストアップした中でスタック位置確定フラグが暫定であるウィンドウ全てを、それらの中でみたスタック順はそのままで、内部データベースの中でスタック位置確定フラグが確定済みであるスタック最上位の通常ウィンドウの直上に位置させ、更に対象ウィンドウを前記リストアップしたウィンドウの中で最上位となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、リストアップしたウィンドウのスタック位置確定フラグを確定済みに更新し、また、内部データベース全体に対してスタック位置情報を更新した上で、S612に進む。

【００７４】

S610のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図９に示す。

【００７５】

図９に示されるように、対象ウィンドウと同じグループ値を持つスタック位置確定フラグが暫定の通常ウィンドウが、スタック位置確定フラグが確定済みの通常ウィンドウ全体のスタック最上位ウィンドウの直上に移され、同一グループ値を持つウィンドウ全体の中で対象ウィンドウのスタック最上位移動がなされているのが解る。

【００７６】

図６において、S611では、S606でリストアップした中でスタック位置確定フラグが暫定であるウィンドウ全てを、それらの中でみたスタック順はそのままで、全体スタックの最下位とし、更に対象ウィンドウをリストアップしたウィンドウの中で最上位となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、リストアップしたウィンドウのスタック位置確定フラグを確定済みに更新し、また、内部データベース全体に対してスタック位置情報を更新した上で、S612に進む。

【００７７】

S611のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図１０に示す。

【００７８】

図１０に示されるように、対象ウィンドウと同じグループ値を持つスタック位置確定フラグが暫定の通常ウィンドウが、スタック最下位に移され、同一グループ値を持つ通常ウィンドウ全体の中で対象ウィンドウのスタック最上位移動がなされているのが解る。

【００７９】

図６において、S612では、処理を終了する。

【００８０】

図１において、アプリケーションプログラム１０１，１０２、１０３は、生成及びウィンドウ種類及びグループ設定したウィンドウに対して、グループ内スタック最下位移動をウィンドウシステム１２に対して発行する事ができる。Xウィンドウシステムの場合、例えば、グループ内スタック最下位移動にXLowerWindow()関数を割り当てれば良い。また、優先ウィンドウに対して、スタック変更要求として、優先ウィンドウスタック最下位移動要求をウィンドウシステム１２に対して発行する事ができ、例えば、グループ内スタック最下位移動要求と同じXLowerWindow()関数を割り当てる。

【００８１】

アプリケーションプログラムがXLowerWindow()を実施した場合、ウィンドウシステム１２はスタック変更をせず、ウィンドウマネージャ１０５に対してスタック最下位移動要求イベント（ConfigureRequest）を発行する。スタック最下位移動要求イベントを受け取った際のウィンドウマネージャ１０５の動作例となるフロー図を図１１に示す。

【００８２】

図１１に示されるように、S1101では、該時点で受け取ったスタック最下位移動要求イベント（ConfigureRequest）から対象ウィンドウの識別子を獲得して、S1102に進む。

【００８３】

S1102では、対象ウィンドウの識別子で内部データベースを検索し、対応するデータが存在する場合はS1103へ進み、存在しない場合はS1112へ進む。

【００８４】

S1103では、内部データベースにあるプロパティ獲得フラグが未獲得である全ウィンドウについて、ウィンドウシステム１２に対してウィンドウ種類及びグループ値を問い合わせ、獲得できたウィンドウについては、そのウィンドウ種類及びグループ値とプロパティ獲得済みである事を内部データベースに記憶し、S1104に進む。

【００８５】

S1104では、対象ウィンドウのウィンドウ種類をチェックし、優先ウィンドウの場合はS1105へ進み、通常ウィンドウの場合はS1106へ進む。

【００８６】

S1105では、内部データベースからウィンドウ種類が優先ウィンドウであるものの内、スタック最下位のウィンドウを検索し、そのウィンドウが対象ウィンドウ自身であれば、何もせずS1112へ進み、そのウィンドウが対象ウィンドウ自身でなければ、そのウィンドウの直下となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、内部データベース全体に対してスタック位置情報を更新した上で、S1112に進む。

【００８７】

S1105のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図１２に示す。

【００８８】

図１２に示されるように、対象ウィンドウのみが優先ウィンドウの中のスタック最下位のウィンドウの直下に移されているのが解る。

【００８９】

図１１において、S1106では、対象ウィンドウと同じグループに属する通常ウィンドウを内部データベースから検索してリストアップし、S1107に進む。

【００９０】

S1107では、前記リストアップした中にスタック位置確定フラグが確定済みであるウィンドウが存在するならばS1108に進み、存在しないならS1109に進む。

【００９１】

S1108では、リストアップした中でスタック位置確定フラグが暫定であるウィ

ンドウがあれば、それら全てを、それらの中でみたスタック順はそのままで、リストアップした中でスタック位置確定フラグが確定済みであるウィンドウの内の最上位のウィンドウの直上に位置させ、更に対象ウィンドウをリストアップしたウィンドウの中で最下位となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、リストアップした中でスタック位置確定フラグが暫定であるウィンドウがあれば確定済みに更新し、また、内部データベース全体に対してスタック位置情報を更新した上で、S1112に進む。

【００９２】

S1108のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図１３に示す。図１３に示されるように、対象ウィンドウと同じグループ値を持つスタック位置確定フラグが暫定の通常ウィンドウが、同じグループ値を持つスタック位置確定フラグが確定済みの通常ウィンドウの直上に移され、同一グループ値を持つ通常ウィンドウ全体の中で対象ウィンドウのスタック最下位移動がなされているのが解る。

【００９３】

図１１において、S1109では、内部データベースにスタック位置確定フラグが確定済みの通常ウィンドウが存在するならばS1110に進み、存在しないならばS1111に進む。

【００９４】

S1110では、S1106でリストアップした中でスタック位置確定フラグが暫定であるウィンドウ全てを、それらの中でみたスタック順はそのままで、内部データベースの中でスタック位置確定フラグが確定済みであるスタック最上位の通常ウィンドウの直上に位置させ、更に対象ウィンドウを前記リストアップしたウィンドウの中で最下位となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、リストアップしたウィンドウのスタック位置確定フラグを確定済みに更新し、また、内部データベース全体に対してスタック位置情報を更新した上で、S1112に進む。

【００９５】

S1110のスタック変更要求によってスタックがどのように変化するかを示す具

体例となる模式図を図１４に示す。

【００９６】

図１４に示されるように、対象ウィンドウと同じグループ値を持つスタック位置確定フラグが暫定の通常ウィンドウが、スタック位置確定フラグが確定済みの通常ウィンドウ全体のスタック最上位ウィンドウの直上に移され、同一グループ値を持つウィンドウ全体の中で対象ウィンドウのスタック最上位移動がなされているのが解る。

【００９７】

図１１において、S1111では、S1106でリストアップした中でスタック位置確定フラグが暫定であるウィンドウ全てを、それらの中でみたスタック順はそのままで、全体スタックの最下位とし、更に対象ウィンドウをリストアップしたウィンドウの中で最上位となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、リストアップしたウィンドウのスタック位置確定フラグを確定済みに更新し、また、内部データベース全体に対してスタック位置情報を更新した上で、S1112に進む。

【００９８】

S1111のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図１５に示す。

【００９９】

図１５に示されるように、対象ウィンドウと同じグループ値を持つスタック位置確定フラグが暫定の通常ウィンドウが、スタック最下位に移され、同一グループ値を持つ通常ウィンドウ全体の中で対象ウィンドウのスタック最上位移動がなされているのが解る。

【０１００】

図１１において、S1112では、処理を終了する。

【０１０１】

図１において、アプリケーションプログラム１０１，１０２、１０３は、生成及びウィンドウ種類及びグループ設定した通常ウィンドウに対して、グループ単位スタック最上位移動をウィンドウマネージャ１０５に対して発行する事ができ

る。Xウィンドウシステムの場合、例えば、クライアントメッセージ機構を利用する事で容易に実現できる。今後、クライアントメッセージでウィンドウマネージャ１０５に対して発行するグループ単位スタック最上位移動要求のメッセージの事をグループ単位スタック最上位移動要求メッセージと呼ぶ事とする。そして、このグループ単位スタック最上位移動要求メッセージには、移動したいグループ値が含まれているものとする。

【０１０２】

アプリケーションプログラムがグループ単位スタック最上位移動要求メッセージ（ClientMessage）を発行した場合、ウィンドウシステム１２経由でウィンドウマネージャ１０５に転送される。　グループ単位スタック最上位移動要求メッセージを受け取った際のウィンドウマネージャ１０５の動作例となるフローを図１６に示す。

【０１０３】

図１６に示されるように、S1601では、該時点で受け取ったグループ単位スタック最上位移動要求メッセージ（ClientMessage）から対象グループ値を獲得して、S1602に進む。

【０１０４】

S1602では、内部データベースにあるプロパティ獲得フラグが未獲得である全ウィンドウについて、ウィンドウシステム１２に対してウィンドウ種類及びグループ値を問い合わせ、獲得できたウィンドウについては、そのウィンドウ種類及びグループ値とプロパティ獲得済みである事を内部データベースに記憶し、S1603に進む。

【０１０５】

S1603では、対象グループ値を持つ通常ウィンドウ情報を内部データベースから検索してリストアップし、S1604に進む。

【０１０６】

S1604では、内部データベースにスタック位置確定情報が確定済み且つグループ値情報が対象グループ値とは異なる通常ウィンドウが存在すればS1605に進み、存在しなければS1606に進む。

【０１０７】

S1605では、リストアップした中でのスタック順はそのままで、内部データベースの中でスタック位置確定フラグが確定済み且つグループ値情報が対象グループ値とは異なる通常ウィンドウの内でスタック最上位のウィンドウの直上となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、リストアップした中でスタック位置確定フラグが暫定であるウィンドウがあれば確定済みに更新し、また、内部データベース全体に対してスタック位置情報を更新した上で、S1607に進む。

【０１０８】

S1605のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図１７に示す。

【０１０９】

図１７に示されるように、対象グループ値を持つ通常ウィンドウが、スタック位置確定フラグが確定済み且つグループ値情報が対象グループ値とは異なる通常ウィンドウ全体のスタック最上位ウィンドウの直上に移されているのが解る。

【０１１０】

図１６におけるS1606では、リストアップした中でのスタック順はそのままで、全体スタックの最下位となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、リストアップした中でスタック位置確定フラグが暫定であるウィンドウがあれば確定済みに更新し、また、内部データベース全体に対してスタック位置情報を更新した上で、S1607に進む。

S1606のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図１８に示す。

【０１１１】

図１８に示されるように、対象グループ値を持つ通常ウィンドウが、全体スタックの最下位に移されているのが解る。

【０１１２】

図１６において、S1607では、処理を終了する。

【０１１３】

図１において、アプリケーションプログラム１０１，１０２、１０３は、生成及びウィンドウ種類及びグループ設定した通常ウィンドウに対して、グループ単位スタック最下位移動をウィンドウマネージャ１０５に対して発行する事ができる。Xウィンドウシステムの場合、例えば、クライアントメッセージ機構を利用する事で容易に実現できる。今後、クライアントメッセージでウィンドウマネージャ１０５に対して発行するグループ単位スタック最下位移動要求のメッセージの事をグループ単位スタック最下位移動要求メッセージと呼ぶ事とする。そして、このグループ単位スタック最下位移動要求メッセージには、移動したいグループ値が含まれているものとする。

【０１１４】

アプリケーションプログラムがグループ単位スタック最下位移動要求メッセージ（ClientMessage）を発行した場合、ウィンドウシステム１２経由でウィンドウマネージャ１０５に転送される。

【０１１５】

グループ単位スタック最下位移動要求メッセージを受け取った際のウィンドウマネージャ１０５の動作例となるフローを図１９に示す。

【０１１６】

図１９に示されるように、S1901では、該時点で受け取ったグループ単位スタック最下位移動要求メッセージ（ClientMessage）から対象グループ値を獲得して、S1902に進む。

【０１１７】

S1902では、内部データベースにあるプロパティ獲得フラグが未獲得である全ウィンドウについて、ウィンドウシステム１２に対してウィンドウ種類及びグループ値を問い合わせ、獲得できたウィンドウについては、そのウィンドウ種類及びグループ値とプロパティ獲得済みである事を内部データベースに記憶し、S1903に進む。

【０１１８】

S1903では、対象グループ値を持つ通常ウィンドウを内部データベースから検索してリストアップし、S1904に進む。

【０１１９】

S1904では、リストアップした中でのスタック順はそのままで、全体スタックの最下位となるよう、ウィンドウシステム１２にスタック変更要求を発行し、次に、リストアップした中でスタック位置確定フラグが暫定であるウィンドウがあれば確定済みに更新し、また、内部データベース全体に対してスタック位置情報を更新した上で、S1905に進む。

【０１２０】

S1904のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図２０に示す。

【０１２１】

図２０に示されるように、対象グループ値を持つ通常ウィンドウが、全体スタックの最下位に移されているのが解る。

【０１２２】

図１９におけるS1905では、処理を終了する。

【０１２３】

図１において、アプリケーションプログラム１０１，１０２、１０３は、生成したウィンドウに対して破棄要求を発行する。

【０１２４】

ウィンドウシステム１２は、破棄要求を受け取ると、該当するウィンドウを管理から外して削除する。そして、ウィンドウマネージャ１０５にウィンドウ破棄通知イベント（DestroyNotify）を発行する。

【０１２５】

ウィンドウマネージャ１０５は、ウィンドウ破棄通知イベント（DestroyNotify）を受け取ると、そこからウィンドウ識別子を抽出して、該当するウィンドウ情報を内部データベースから検索し削除した後、内部データベース全体に対して必要に応じてスタック位置を更新し、処理を終了する。

【０１２６】

以上のことにより、各通常ウィンドウはそのウィンドウの表示までに、ウィンドウのスタック順がグループ毎に纏められるので、グループ最上位のウィンドウ

は、他のグループのウィンドウの下に重ねられて表示される事が無くなる。また、ウィンドウ管理プログラムが、スタック変更要求を受け取る事を切っ掛けにして、スタック変更対象となるグループに属する通常ウィンドウ全てを、その中でのスタック順は保持したままで纏めた上でスタック変更を実施するので、アプリケーションプログラムは、自グループのウィンドウのスタック順を把握し易くなる。また、ウィンドウ管理プログラムにグループ内のスタック最上位移動とグループ内のスタック最下位移動の機能を設ける事で、グループ内で通常ウィンドウのスタック順を操作できるので、各アプリケーションプログラムは、他のアプリケーションプログラムの通常ウィンドウ表示を気にする事無く、スタック変更を行う事ができる。また、ウィンドウ管理プログラムにグループ単位のスタック最上位移動と最下位移動の機能を設ける事で、例えば、グループをアプリケーションプログラム単位で振り分けた場合だと、そのアプリケーションに属するウィンドウのメンバーを知らなくても、スタック最前面にそのアプリケーションに属する全通常ウィンドウを移動させる事が可能となる。また、優先ウィンドウを設ける事で、グループ単位のスタック順に関係無く、最優先で表示するウィンドウのサービスを提供することができる。

【０１２７】

なお、本実施の形態では、ウィンドウマネージャのウィンドウのウィンドウ種類及びグループ値取得のタイミングをマップ要求イベント(MapRequest)受信時及びスタック変更要求受信時としたが、これに限定されるものではなく、アプリケーションプログラムがウィンドウ種類及びグループ設定をしたタイミングで取得して内部データベースに記憶しておいても良く、その方がむしろ処理効率が良い。更に、このタイミングで、スタック順をグループ毎に纏めても構わない。また、ウィンドウ種類及びグループ値の変更を認めても良いし、認めなくても良い。認める場合は、ウィンドウ種類及びグループ設定のタイミングで、ウィンドウ種類が優先ウィンドウから通常ウィンドウに変更された場合は、スタック位置確定フラグを暫定に変更する。逆にウィンドウ種類が通常ウィンドウから優先ウィンドウに変更された場合は、該時点のスタック最上位に該ウィンドウを位置させるようにスタック変更を行う。また、ウィンドウ種類に変更が無く、グループ値が

変更となった場合は、スタック位置確定フラグを暫定に変更する。そして、内部データベースを更新すれ場良い。また、認めない場合は、ウィンドウ種類及びグループ設定のタイミングで、プロパティ獲得フラグをチェックして、獲得済みであれば、該設定を無視すれば良い。

【０１２８】

また、ウィンドウ種類として優先ウィンドウと通常ウィンドウの２種類を制御する例を示したが、通常ウィンドウの１種類だけであっても容易に実施できる事は言うまでも無い。

【０１２９】

また、ウィンドウマネージャがルートウィンドウの子ウィンドウであるトップレベルウィンドウのスタック順を制御する場合を例に示したが、トップレベルウィンドウの子孫ウィンドウに対しても実施する事ができる事は言うまでも無い。

【０１３０】

また、ウィンドウ管理プログラムをXウィンドウシステムとウィンドウマネージャとして、アプリケーションプログラムとのインターフェースにウィンドウ生成要求、マップ要求を用いたが、これに限定されるものではなく、表示要求に相当するインターフェースさえあれば良く、その場合、ウィンドウ管理プログラムは、ウィンドウ表示要求受信時に、本実施の形態で示したウィンドウ生成要求受信時の処理とマップ要求受信時の処理を纏めて行えば良い。

【０１３１】

また、ウィンドウのスタック順をグループ毎に一連とする際、単にウィンドウのスタック順を変更するだけであったが、例えば、ウィンドウ管理プログラムが、グループ毎に代表ウィンドウを１枚生成し、ウィンドウのスタック順をグループ毎に一連とする際に、前記ウィンドウを前記代表ウィンドウの子ウィンドウとするようにして制御することも容易である。そうすることによって、グループ単位のスタック移動やグループ単位のウィンドウの表示/非表示の切り替えを代表ウィンドウに対する制御で実施できる分、グループ内のウィンドウが多いほど、処理負荷を軽減することができる。代表ウィンドウ使用下でのスタックの様子を示す例となる模式図を図２５に示す。

【０１３２】

また、最上位グループよりスタック下位のウィンドウ表示について、ウィンドウ管理プログラムでは非表示とするような制御はしていないが、非表示とする制御を行う事も容易に実現できる。その場合、ウィンドウ情報に表示/非表示を示すマップフラグを追加して、アプリケーションプログラムの表示要求状態を記憶しておき、該ウィンドウの所属するグループが通常ウィンドウの中でスタック最上位となった時、前記マップフラグをチェックし、表示となっていればマップ処理を行い、非表示となっていればアンマップのままとする。逆に、該ウィンドウの所属するグループが通常ウィンドウの中でスタック最上位でなくなった際には、前記マップフラグが表示となっていればアンマップ処理を行い、非表示となっていれば何もしない。これにより、通常ウィンドウの中の最上位グループよりスタック下位のグループが何を表示しようとしても，表示装置に表示されないので、最上位グループのウィンドウで覆われない部分に中途半端に映り込む心配が無い。

【０１３３】

また、通常ウィンドウの最上位グループに属する全ウィンドウの中のスタック最下位のウィンドウの直下に、ウィンドウ管理プログラムが生成及びマップする特定のウィンドウを位置させなかったが、位置させることも容易に実現できる。これにより、例えば、前記特定のウィンドウを表示装置のスクリーン一杯に表示しておけば、前記特定のウィンドウのスタック下位のグループが何を表示しようとしても，表示装置に表示されないので、最上位グループのウィンドウで覆われない部分に中途半端に映り込む心配が無い。前記特定のウィンドウを使用した場合のスタックの様子を示す例となる模式図を図２６に示し、また、スクリーンに表示されている例となる模式図を図２７に示す。

【０１３４】

また、アプリケーションプログラムが３つの場合を例に示したが、これに限定されるものではなく、１つ以上であれば良い。

【０１３５】

また、各アプリケーションプログラムに１つのグループを割り当てた場合を例

に示したが、これに限定されるものではなく、１つのアプリケーションに複数のグループを割り当てても良く、また、複数のアプリケーションに１つのグループを割り当てても構わない。

【０１３６】

また、ウィンドウ管理プログラムをXウィンドウシステムとウィンドウマネージャとして、両者間でイベントや各種要求を送受信し合う例を示したが、これに限定されるものではなく、Xウィンドウシステムである必要は無いし、また、２つに分けてイベントや各種要求を送受信する構成である必要もなく、一体化して実施する場合は、イベントやメッセージのインターフェースを関数ベースに変える事で本実施の形態を容易に応用できる。

【０１３７】

また、通常ウィンドウの最上位グループに属する全ウィンドウの中のスタック最上位のウィンドウの直上に、ウィンドウ管理プログラムが生成する特定のウィンドウを位置させなかったが、位置させることも容易に実現できる。これにより、最上位グループを別のグループに入れ替える際、Xウィンドウシステムの場合、XRestackWindows（R）関数を使用する事となるが、ウィンドウ群のスタック移動先を示すために使用できる。もし、最上位グループのウィンドウよりスタック上位にウィンドウが存在しなかった場合、存在する全てのウィンドウの並び順を決め、前記XRestackWindows（R）の引数に代入せねばならないが、前記特定のウィンドウがあれば、前記特定のウィンドウと前記ウィンドウ群を引数に代入するだけで済む。

【０１３８】

また、Xウィンドウシステムを用いる場合、ウィンドウマネージャは、スタックを操作するためにXRestackWindows（R）関数を使用する。XRestackWindows（R）関数の引数には、ウィンドウ群とそのウィンドウ群の移動先を示すウィンドウを指定するが、それらの何れかが破棄されていた場合、スタック操作が意図した通りに実施されない事がある。そのための対策として、例えば、ウィンドウマネージャは、ウィンドウ破棄通知イベント（DestroyNotify）を受け取った際に、該当するウィンドウ情報を内部データベースから検索し削除し、内部データベ

ース全体に対して必要に応じてスタック位置を更新した後、次に、ウィンドウシステムの該時点でのスタックツリーを問い合わせて獲得し、ウィンドウマネージャが認識していない新規のウィンドウは対象から外し、ウィンドウマネージャのスタックツリーとの整合性をチェックし、ウィンドウマネージャの認識と異なっている場合は、ウィンドウマネージャのスタックツリーに合わせ込むスタック変更要求を発行し、認識が合うまで、ウィンドウシステムの該時点でのスタックツリー問い合わせ、整合性チェック、スタック変更要求の発行を繰り返すようにしても良い。これにより、ウィンドウシステムがスタック変更処理に失敗しても、ウィンドウシステム間とのスタック認識の整合性を確保でき、処理が破綻する事がなくなる。

【０１３９】

また、ウィンドウマネージャが、XRestackWindows（Ｒ）を実行した際にフラグを立て、ウィンドウ破棄通知イベント（DestroyNotify）を受け取った際に、前記フラグが立っている場合にのみ、前記フラグを下げると共にスタックツリーの整合性のための処理を行うようにしても良い。これにより、スタックツリーの整合性のための処理の実行頻度を軽減できる。

【０１４０】

また、コンピュータ１とは、家庭用コンピュータや携帯電話、PDA（Personal Digital Assistance）、セットトップボックス、デジタルスチルカメラ、カメラ一体型VTR等、プログラムが動作するためのCPUとメモリを実装して表示装置に複数のウィンドウを表示しようとする装置であれば何でも良い事は言うまでもない。

【０１４１】

また、表示装置１１をコンピュータ１の一部としたが、物理上で一体化している事を指すものではなく、コンピュータ１と有線/無線を問わず、何らかの形で接続されていさえすれば良い。

【０１４２】

また、本実施の形態のウィンドウスタック制御方法を内臓するプログラムは、ROMやディスク等に記憶して供給する事ができるほか、ネットワークを介して供

給する事が可能である事は言うまでもない。

【０１４３】

（実施の形態２）

本実施の形態のウィンドウスタック制御方法は、実施の形態１を変形したものであり、実施の形態１と実施の形態２の違いは、マップ要求イベント(MapRequest)を受け取った際のウィンドウマネージャ１０５の動作と、アプリケーションプログラムがウィンドウ種類及びグループ値を設定するタイミングでウィンドウマネージャ１０５はスタック順をグループ毎に纏めてはいけないことだけである。よって、本実施例では、ウィンドウマネージャ１０５のマップ要求イベント(MapRequest)を受け取った際のウィンドウマネージャ１０５の動作についてのみ説明する。

【０１４４】

マップ要求イベント(MapRequest)を受け取った際のウィンドウマネージャ１０５の動作例となるフロー図を図２１に示す。

【０１４５】

図２１に示されるように、S2101では、該時点で受け取ったマップ要求イベント(MapRequest)からマップ対象ウィンドウの識別子を獲得して、S2102に進む。

【０１４６】

S2102では、マップ対象ウィンドウの識別子で内部データベースを検索し、対応するウィンドウ情報が存在する場合はS2103へ進み、存在しない場合はS2115へ進む。

【０１４７】

S2103では、内部データベースで対象ウィンドウのスタック位置確定フラグが確定済みかどうかをチェックし、確定済みの場合はS2114に進み、暫定の場合はS2104に進む。

【０１４８】

S2104では、対象ウィンドウのプロパティ獲得フラグをチェックし、獲得済みの場合はS2106に進み、未獲得の場合はS2105に進む。

【０１４９】

S2105では、対象ウィンドウのウィンドウ種類及びグループ値をウィンドウシステムに問い合わせて獲得し、内部データベースに格納し、S2106に進む。

【０１５０】

S2106では、対象ウィンドウのウィンドウ種類をチェックし、優先ウィンドウの場合は、S2114に進み、通常ウィンドウの場合は、S2107に進む。

【０１５１】

S2107では、内部データベースにあるプロパティ獲得フラグが未獲得である全ウィンドウについて、ウィンドウシステム１２に対してウィンドウ種類及びグループ値を問い合わせ、獲得できたウィンドウについては、そのウィンドウ種類及びグループ値とプロパティ獲得済みである事を内部データベースに記憶し、S2108に進む。

【０１５２】

S2108では、対象ウィンドウと同じグループに属する通常ウィンドウを内部データベースから検索してリストアップし、S2109に進む。　S2109では、前記リストアップした中にスタック位置確定フラグが確定済みであるウィンドウが存在するならばS2110に進み、存在しないならS2111に進む。

【０１５３】

S2110では、リストアップした中でスタック位置確定フラグが暫定であるウィンドウ全てを、それらの中でみたスタック順はそのままで、リストアップした中でスタック位置確定フラグが確定済みであるウィンドウの内の最上位のウィンドウの直上となるよう、ウィンドウシステム１２にスタック変更要求を発行し、S2114に進む。　S2110のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図２２に示す。

【０１５４】

図２２に示されるように、マップ対象ウィンドウ及びマップ対象ウィンドウと同じグループ値を持つスタック位置確定フラグが暫定の通常ウィンドウが、前記グループ値を持つスタック位置確定フラグが確定済みの通常ウィンドウの直上に移されているのが解かる。

【０１５５】

図２１において、S2111では、内部データベースにスタック位置確定フラグが確定済みの通常ウィンドウが存在するならばS2112に進み、存在しないならばS2113に進む。

【０１５６】

S2112では、S2108でリストアップした中でスタック位置確定フラグが暫定である通常ウィンドウ全てを、それらの中でみたスタック順はそのままで、内部データベースの中でスタック位置確定フラグが確定済みであるスタック最上位の通常ウィンドウの直上となるよう、ウィンドウシステム１２にスタック変更要求を発行し、S2114に進む。

【０１５７】

S2112のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図２３に示す。　図２３に示されるように、マップ対象ウィンドウ及びマップ対象ウィンドウと同じグループ値を持つスタック位置確定フラグが暫定の通常ウィンドウが、スタック位置確定フラグが確定済みの通常ウィンドウ全体のスタック最上位ウィンドウの直上に移されているのが解る。

【０１５８】

図２１において、S2113では、S2108でリストアップした中でスタック位置確定フラグが暫定である通常ウィンドウ全てを、それらの中でみたスタック順はそのままで、全体スタックの最下位となるよう、ウィンドウシステム１２にスタック変更要求を発行し、S2114に進む。

【０１５９】

S2113のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図を図２４に示す。

【０１６０】

図２４に示されるように、マップ対象ウィンドウ及びマップ対象ウィンドウと同じグループ値を持つスタック位置確定フラグが暫定の通常ウィンドウが、スタック最下位に移されているのが解る。

【０１６１】

図２１において、S2114では、ウィンドウシステム１２に対してマップ対象ウ

ィンドウのマップ要求を発行し、内部データベースの対象ウィンドウのウィンドウ情報の内、スタック位置確定フラグが未確定を示している場合は確定済みに更新し、また、内部データベース全体に対して必要に応じてスタック位置を更新した上で、S2115に進む。

【０１６２】

S2115では、処理を終了する。

【０１６３】

以上のことにより、ウィンドウ管理プログラムが、マップ対象ウィンドウと同じグループ値を持つスタック位置が暫定の通常ウィンドウ群を、その通常ウィンドウ群の中でのスタック上下関係は保持しつつスタック位置を移してグループ毎に纏めるため、マップによって、同一グループ内でみたスタック順は変更されず、アプリケーションの同一グループ内のスタック順管理の負担が軽減される。更に、実施の形態１で示したように、スタック変更要求受信時にも、対象ウィンドウと同じグループ値を持つスタック位置が暫定の通常ウィンドウ群を、その通常ウィンドウ群の中でのスタック上下関係は保持しつつグループ毎に纏める事で、アプリケーションプログラムは、自グループに属するウィンドウのスタック順をウィンドウの生成順を基本として管理する事ができる。これは、Xウィンドウシステムの基本と一致しており、アプリケーションプログラムは、自グループの通常ウィンドウについて、一般的なスタック順決定ルールである生成順ベースで重なりを管理すれば良くなる。

【０１６４】

なお、本実施の形態では、マップ要求イベント(MapRequest)受信時に、ウィンドウマネージャのウィンドウのグループ値取得を行っているが、これに限定されるものではなく、実施の形態１と同様、アプリケーションプログラムがウィンドウ種類及びグループ設定をしたタイミングでグループ値を取得して内部データベースにウィンドウ種類とグループ値とプロパティ獲得済みである事を記憶しておいても良く、その方がむしろ処理効率が良い。しかし、実施の形態１と違い、前記タイミングで、スタック順をグループ毎に纏めてはいけない。

【０１６５】

【発明の効果】

以上のように本発明によれば、ウィンドウ制御方法において、アプリケーションプログラムがウィンドウにグループを指定し、ウィンドウ管理プログラムがウィンドウのスタック順をグループ毎に一連とすることにより、グループ最上位のウィンドウは、他のグループのウィンドウの下に重ねられる事が無くなり、最前面でウィンドウを表示中のアプリケーションのウィンドウ表示に、他のアプリケーションが不用意に影響を及ぼす事を無くす事ができる。特に、画面が小さく、結果として、ウィンドウの重なり易い携帯電話やPDA等で、その効果は顕著である。

【図面の簡単な説明】

【図１】

本発明の実施の形態１のウィンドウスタック制御方法を実現するシステムの構成例を示す図

【図２】

本発明実施の形態１のマップ要求イベント(MapRequest)を受け取った際のウィンドウマネージャの動作例となるフロー図

【図３】

本発明の実施の形態１の図２のS208のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図４】

本発明の実施の形態１の図２のS210のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図５】

本発明の実施の形態１の図２のS211のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図６】

本発明の実施の形態１のスタック最上位移動要求イベントを受け取った際のウィンドウマネージャの動作例となるフロー図

【図７】

本発明の実施の形態１の図６のS605のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図８】

本発明の実施の形態１の図６のS608のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図９】

本発明の実施の形態１の図６のS610のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図１０】

本発明の実施の形態１の図６のS611のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図１１】

本発明の実施の形態１のスタック最下位移動要求イベントを受け取った際のウィンドウマネージャの動作例となるフロー図

【図１２】

本発明の実施の形態１の図１１のS1105のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図１３】

本発明の実施の形態１の図１１のS1108のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図１４】

本発明の実施の形態１の図１１のS1110のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図１５】

本発明の実施の形態１の図１１のS1111のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図１６】

本発明の実施の形態１のグループ単位スタック最上位移動要求メッセージを受け取った際のウィンドウマネージャの動作例となるフロー図

【図１７】

本発明の実施の形態１の図１６のS1605のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図１８】

本発明の実施の形態１の図１６のS1606のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図１９】

本発明の実施の形態１のグループ単位スタック最下位移動要求メッセージを受け取った際のウィンドウマネージャの動作例となるフロー図

【図２０】

本発明の実施の形態１の図１９のS1904のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図２１】

本発明の実施の形態２のマップ要求イベント(MapRequest)を受け取った際のウィンドウマネージャの動作例となるフロー図

【図２２】

本発明の実施の形態２の図２１のS2110のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図２３】

本発明の実施の形態２の図２１のS2112のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図２４】

本発明の実施の形態２の図２１のS2113のスタック変更要求によってスタックがどのように変化するかを示す具体例となる模式図

【図２５】

本発明の実施の形態１の代表ウィンドウ使用下でのスタックの様子を示す例となる模式図

【図２６】

本発明の実施の形態１の最上位グループに属する全ウィンドウの中のスタック

最下位のウィンドウの直下に、ウィンドウ管理プログラムが生成及びマップする特定のウィンドウを位置させて使用した場合のスタックの様子を示す例となる模式図

【図２７】

本発明の実施の形態１の最上位グループに属する全ウィンドウの中のスタック最下位のウィンドウの直下に、ウィンドウ管理プログラムが生成及びマップする特定のウィンドウを位置させて使用した場合のスクリーンに表示されている例となる模式図

【図２８】

従来例におけるコンピュータ上で動作する３種類のアプリケーションが合計５枚のウィンドウを表示装置であるスクリーンに表示する際の表示例となる模式図

【図２９】

従来例におけるウィンドウスタック順の例となる模式図

【図３０】

従来例におけるウィンドウスタック順の例となる模式図

【符号の説明】

１　スクリーン

７　コンピュータ

８　CPU (Central Processing Unit)

９　メモリ

１０　バス

１１　表示装置

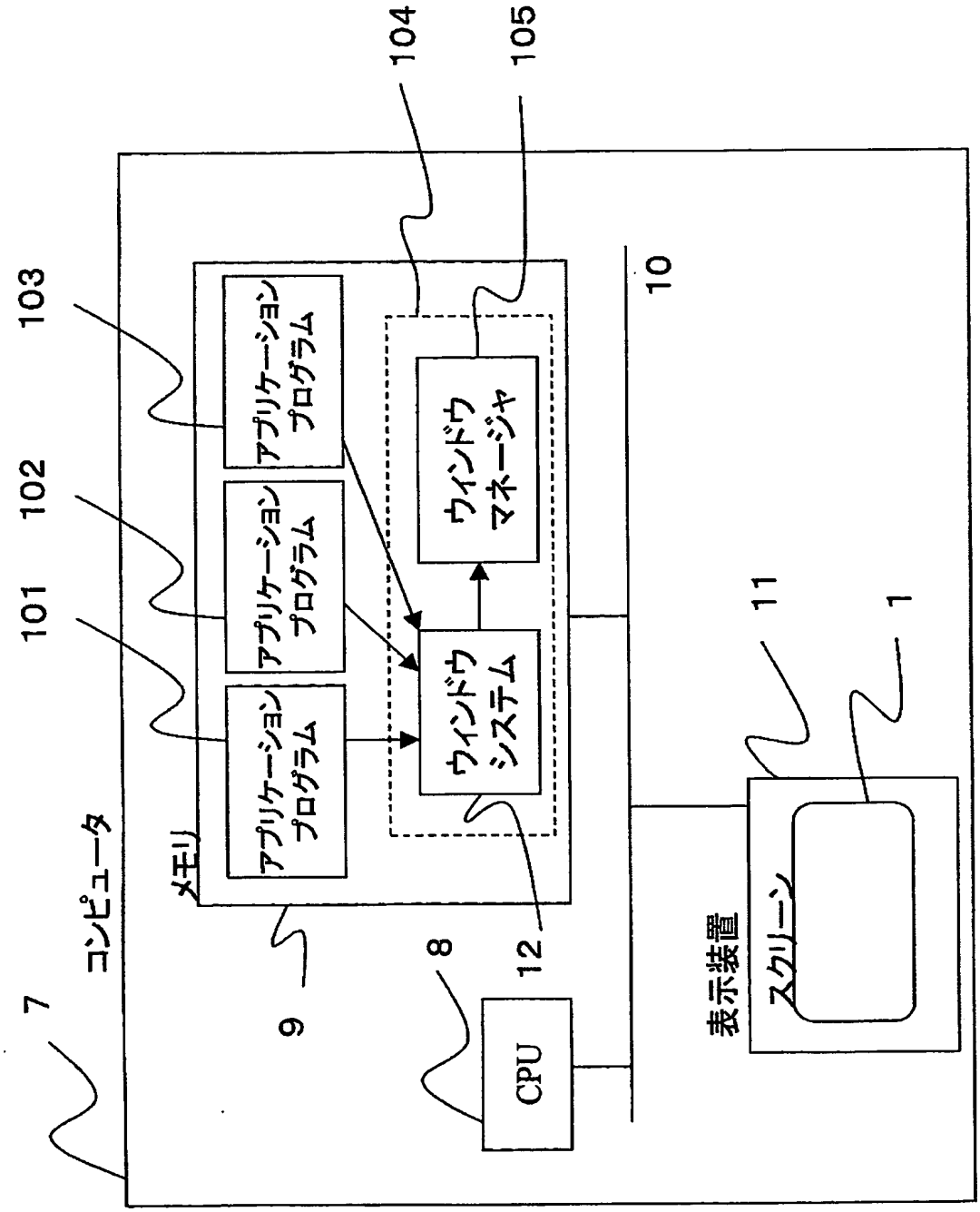
１０１、１０２、１０３　アプリケーションプログラム

１０４　ウィンドウ管理プログラム

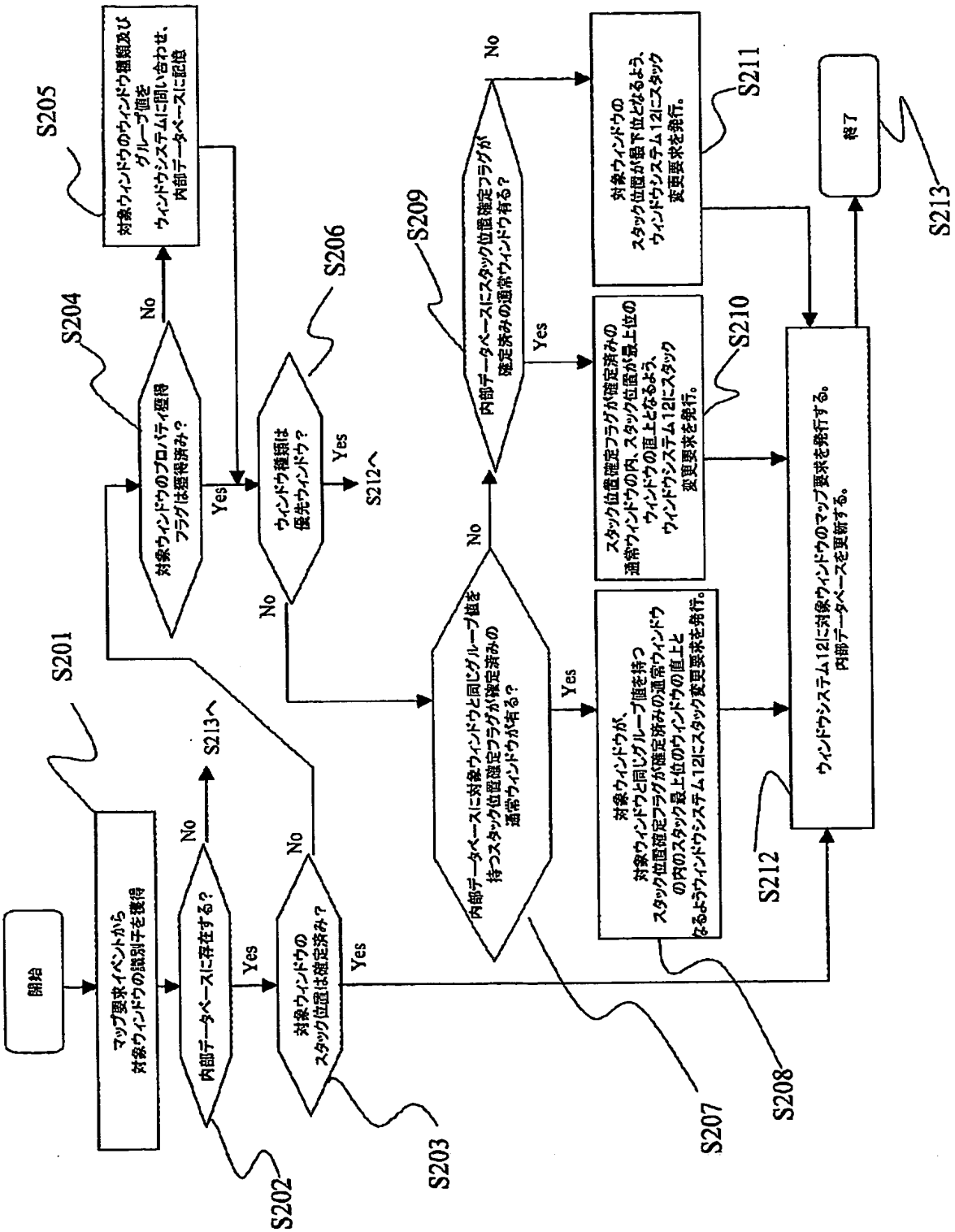
１０５　ウィンドウマネージャ

【書類名】　図面

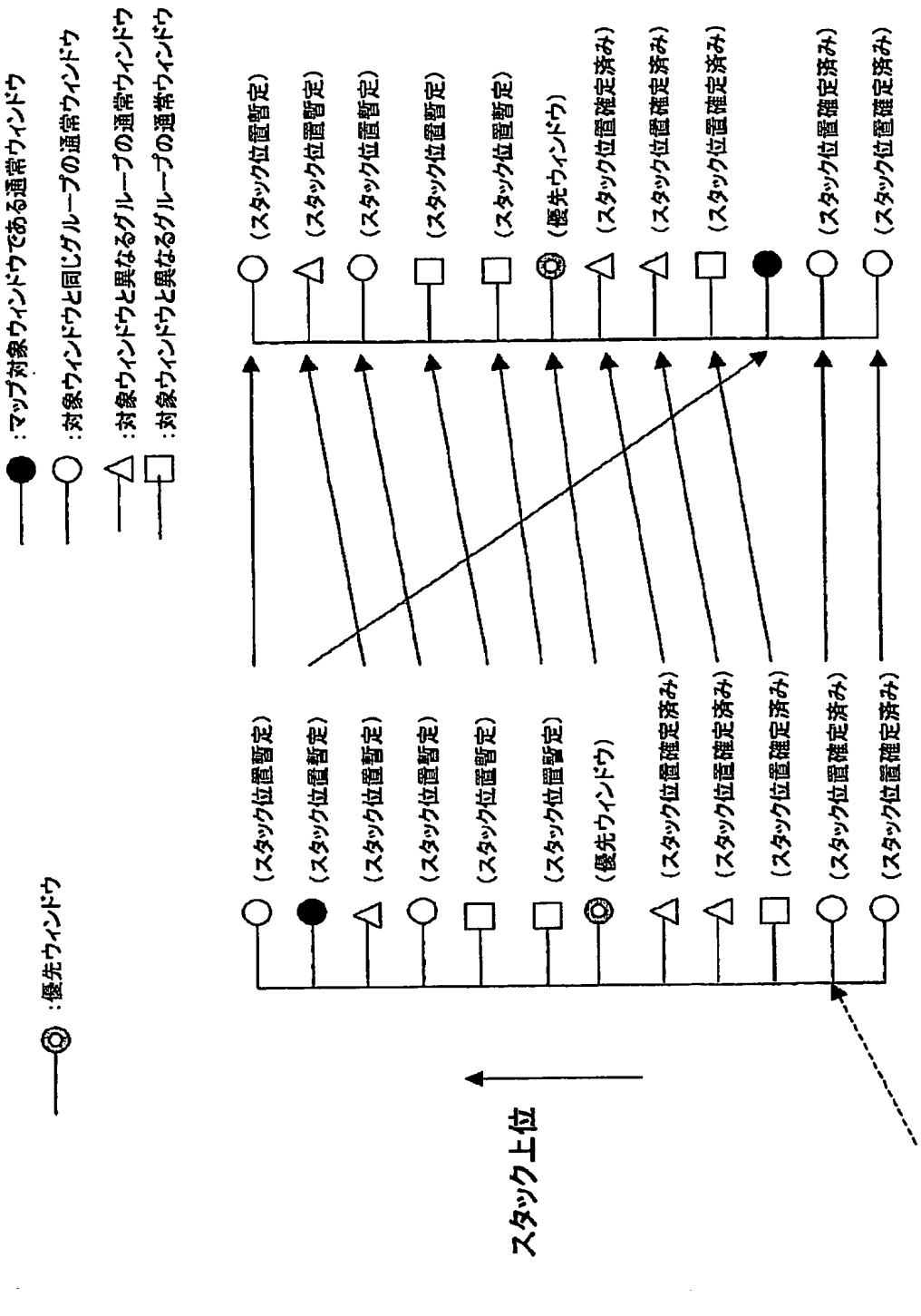
【図１】


104
105
103
102
101
10
アプリケーション
プログラム
アプリケーション
プログラム
アプリケーション
プログラム
ウィンドウ
マネージャ
ウィンドウ
システム
メモリ
9
コンピュータ
7
8
12
CPU
11
1
表示装置
スクリーン

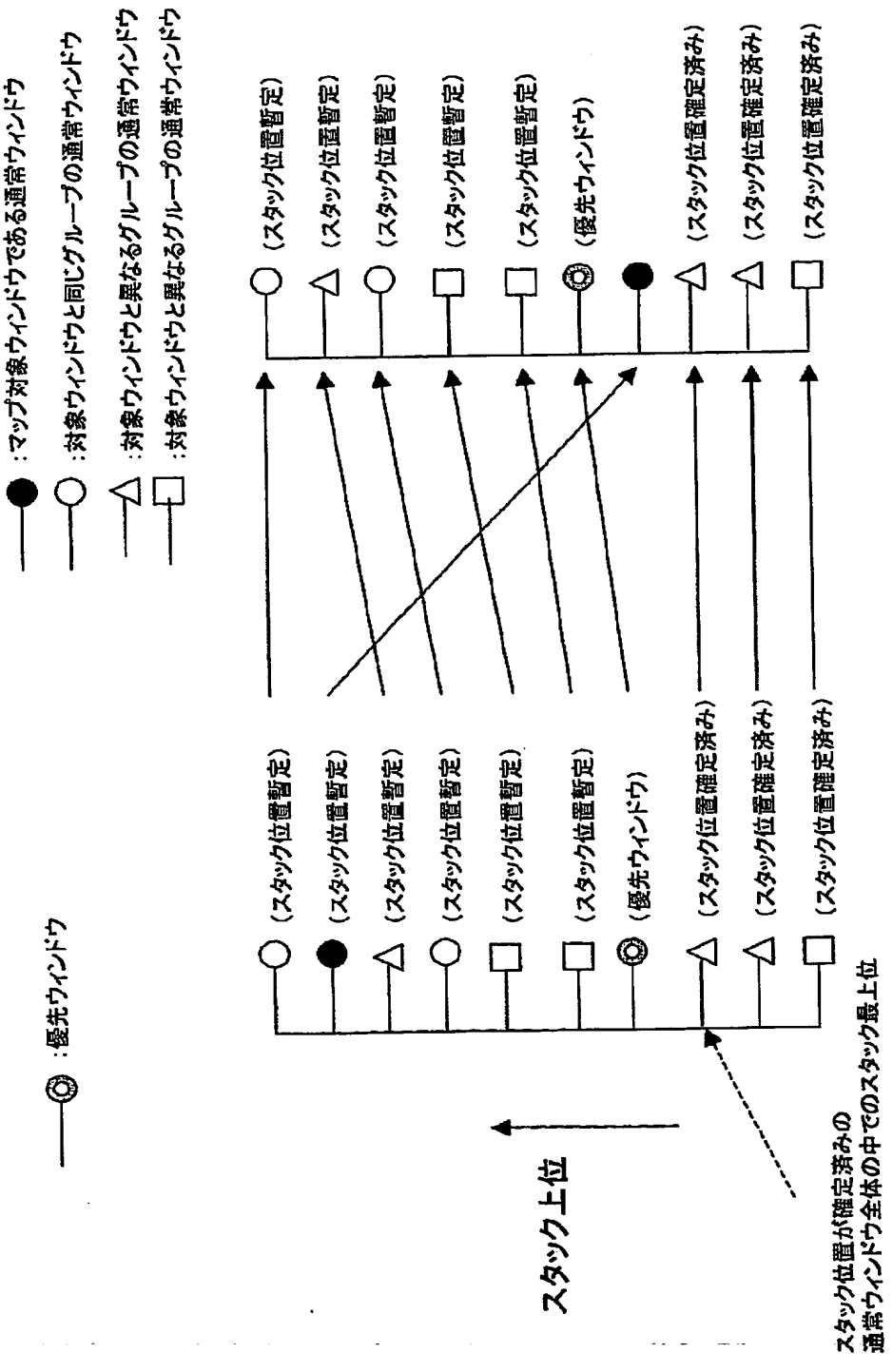
【図２】


開始
マップ要求イベントから
対象ウィンドウの識別子を獲得
S201
内部データベースに存在する？
S202
No
Yes
S213へ
対象ウィンドウの
スタック位置は確定済み？
S203
No
Yes
対象ウィンドウのプロパティ値得
フラグは獲得済み？
S204
No
Yes
対象ウィンドウのウィンドウ種類及び
グループ値を
ウィンドウシステムに問い合わせ、
内部データベースに記憶
S205
ウィンドウ種類は
優先ウィンドウ？
S206
No
Yes
S212へ
内部データベースに対象ウィンドウと同じグループ値を
持つスタック位置確定フラグが確定済みの
通常ウィンドウが有る？
S207
No
Yes
内部データベースにスタック位置確定フラグが
確定済みの通常ウィンドウ有る？
S209
No
Yes
対象ウィンドウが、
対象ウィンドウと同じグループ値を持つ
スタック位置確定フラグが確定済みの通常ウィンドウ
の内のスタック最上位のウィンドウの直上と
なるようウィンドウシステム12にスタック変更要求を発行。
S208
スタック位置確定フラグが確定済みの
通常ウィンドウの内、スタック位置が最上位の
ウィンドウの直上となるよう、
ウィンドウシステム12にスタック
変更要求を発行。
S210
対象ウィンドウの
スタック位置が最下位となるよう、
ウィンドウシステム12にスタック
変更要求を発行。
S211
ウィンドウシステム12に対象ウィンドウのマップ要求を発行する。
内部データベースを更新する。
S212
終了
S213

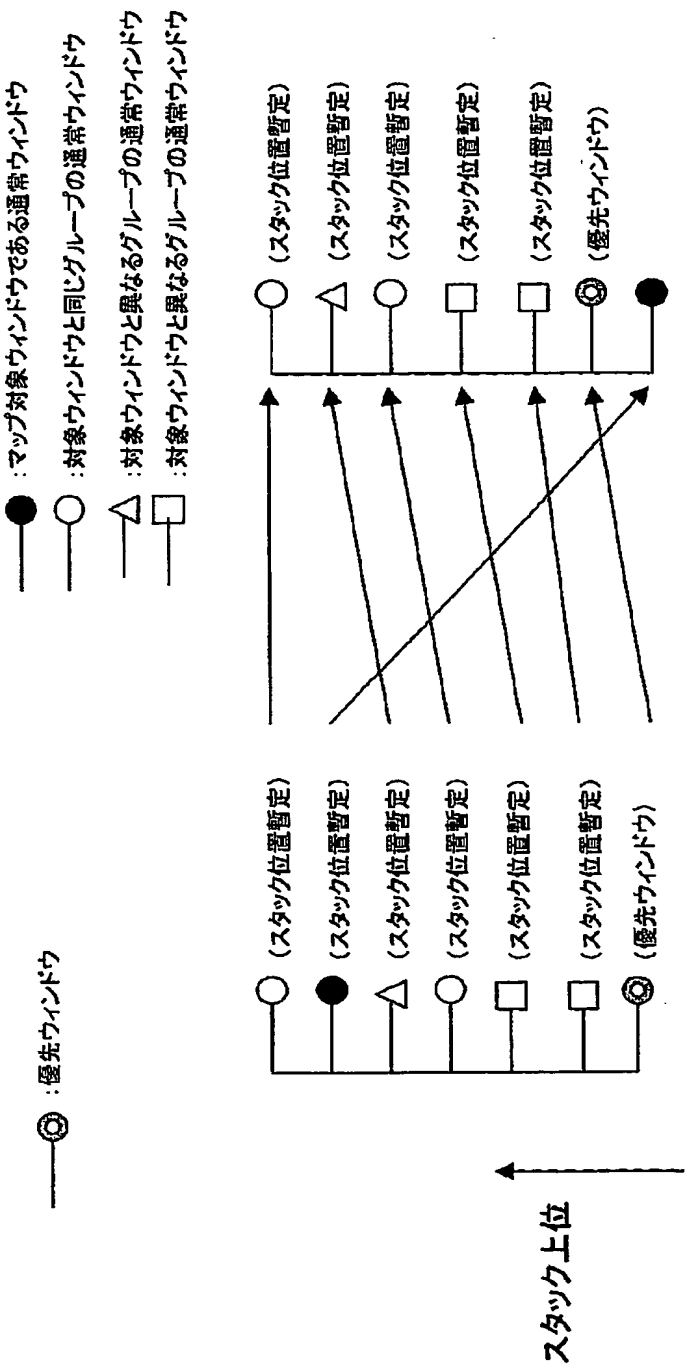
【図３】


:マップ対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
スタック上位
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更前
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更後
同一グループ且つスタック位置が確定済みの
通常ウィンドウの中でのスタック最上位

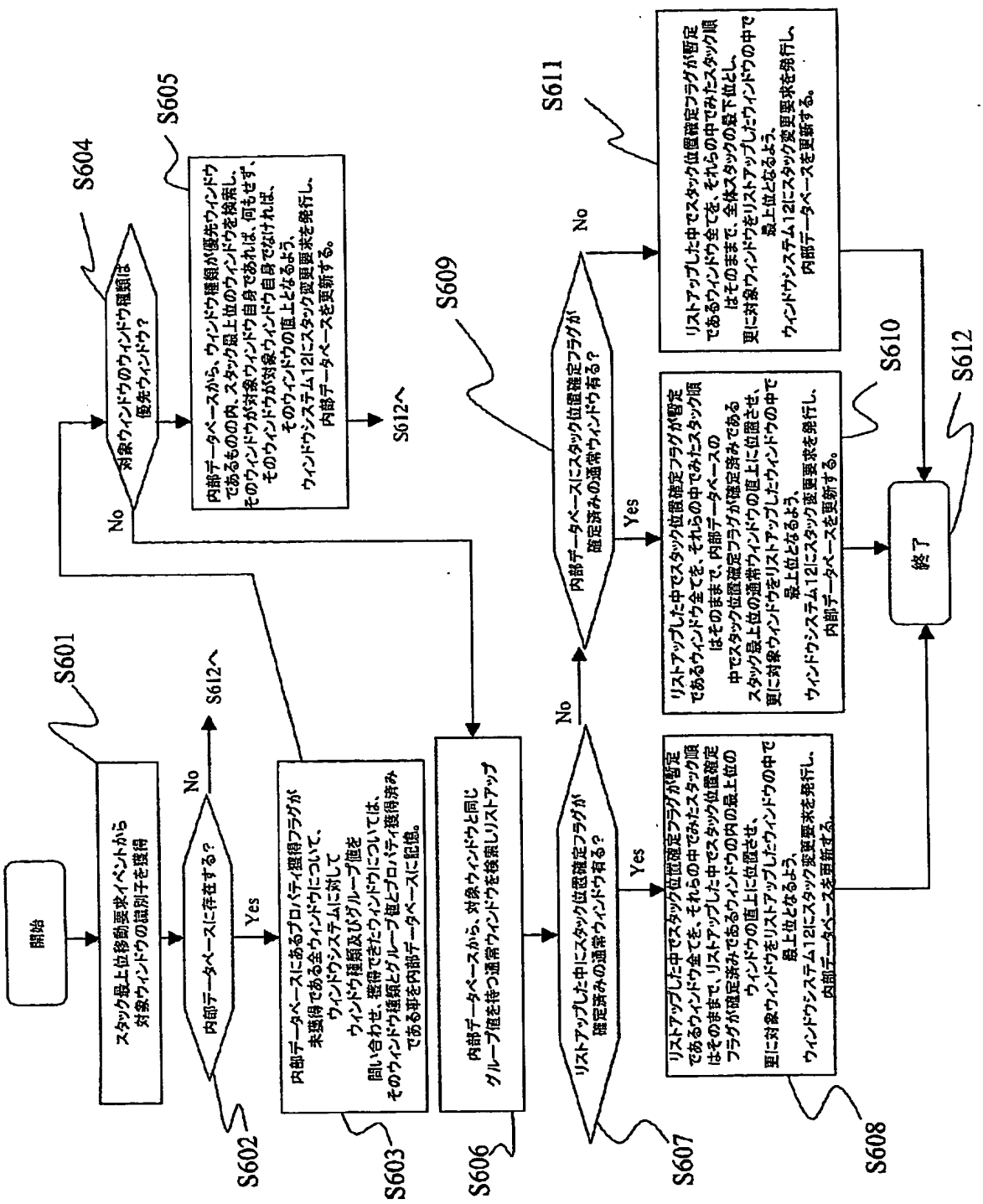
【図４】


:マップ対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
スタック上位
スタック位置が確定済みの
通常ウィンドウ全体の中でのスタック最上位
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更前
スタック変更後

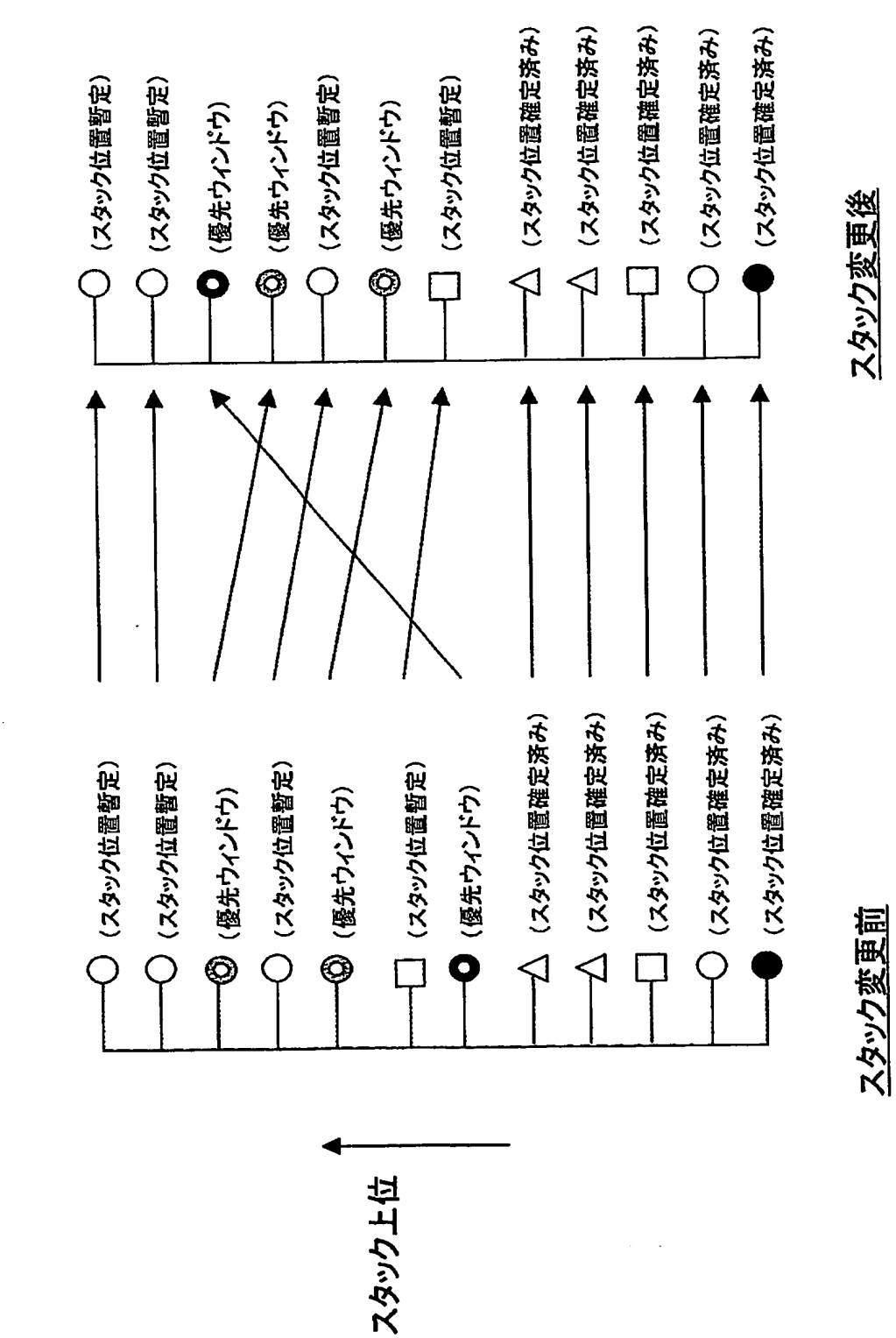
【図５】


:マップ対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
スタック上位
スタック変更前
スタック変更後

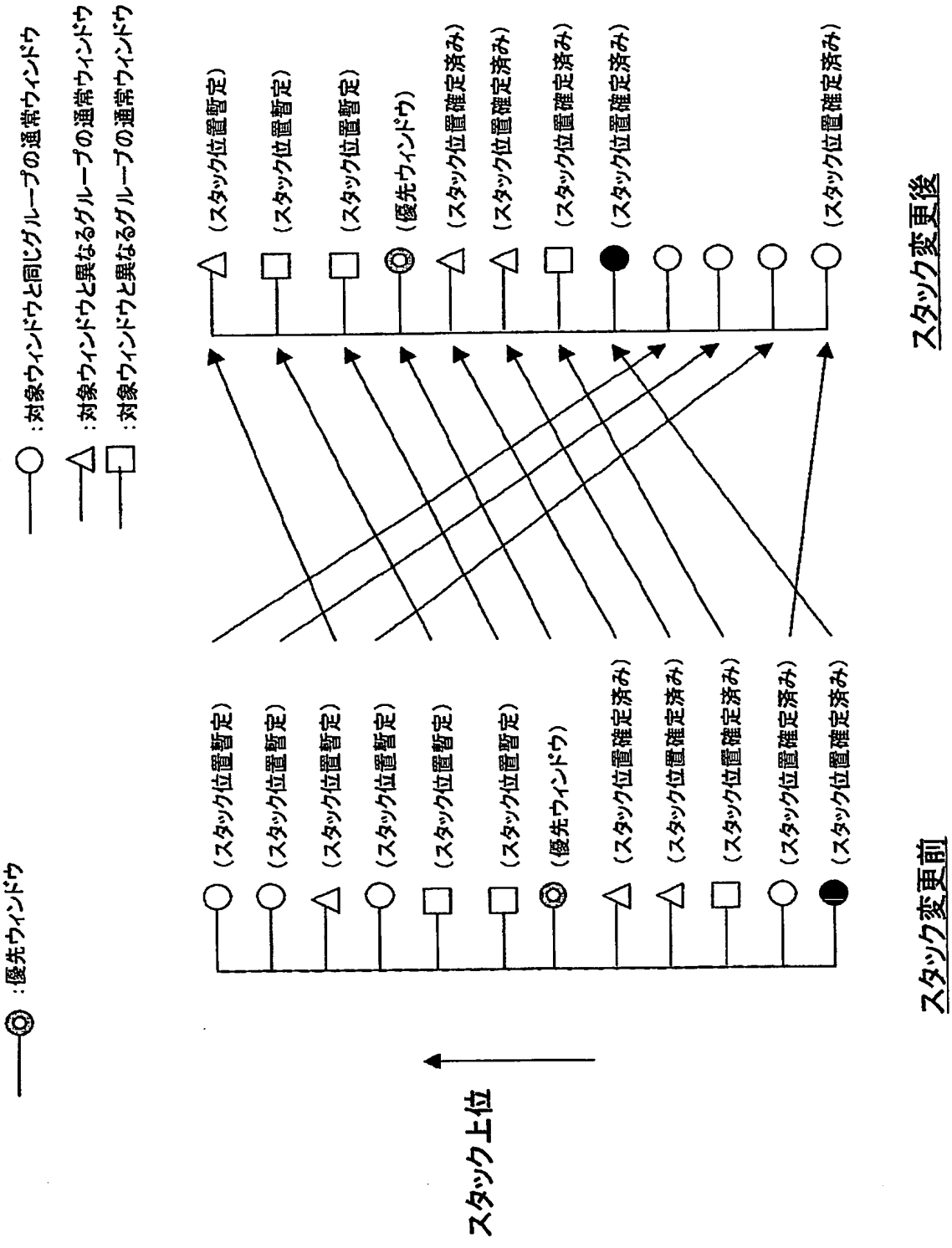
【図６】


開始
スタック最上位移動要求イベントから
対象ウィンドウの識別子を獲得
S601
内部データベースに存在する?
Yes
No
S612へ
S602
内部データベースにあるプロパティ獲得フラグが
未獲得である全ウィンドウについて、
ウィンドウシステムに対して
ウィンドウ種類及びグループ値を
問い合わせ、獲得できたウィンドウについては、
そのウィンドウ種類とグループ値とプロパティ獲得済み
である事を内部データベースに記憶。
S603
内部データベースから、対象ウィンドウと同じ
グループ値を持つ通常ウィンドウを検索しリストアップ
S606
リストアップした中にスタック位置確定フラグが
確定済みの通常ウィンドウ有る?
Yes
No
S607
リストアップした中でスタック位置確定フラグが暫定
であるウィンドウ全てを、それらの中でみたスタック順
はそのままで、リストアップした中でスタック位置確定
フラグが確定済みであるウィンドウの内の最上位の
ウィンドウの直上に位置させ、
更に対象ウィンドウをリストアップしたウィンドウの中で
最上位となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
S608
内部データベースにスタック位置確定フラグが
確定済みの通常ウィンドウ有る?
Yes
No
S609
リストアップした中でスタック位置確定フラグが暫定
であるウィンドウ全てを、それらの中でみたスタック順
はそのままで、内部データベースの
中でスタック位置確定フラグが確定済みである
スタック最上位の通常ウィンドウの直上に位置させ、
更に対象ウィンドウをリストアップしたウィンドウの中で
最上位となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
S610
リストアップした中でスタック位置確定フラグが暫定
であるウィンドウ全てを、それらの中でみたスタック順
はそのままで、全体スタックの最下位とし、
更に対象ウィンドウをリストアップしたウィンドウの中で
最上位となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
S611
終了
S612
対象ウィンドウのウィンドウ種類は
優先ウィンドウ?
S604
No
内部データベースから、ウィンドウ種類が優先ウィンドウ
であるものの内、スタック最上位のウィンドウを検索し、
そのウィンドウが対象ウィンドウ自身であれば、何もせず、
そのウィンドウが対象ウィンドウ自身でなければ、
そのウィンドウの直上となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
S605
S612へ

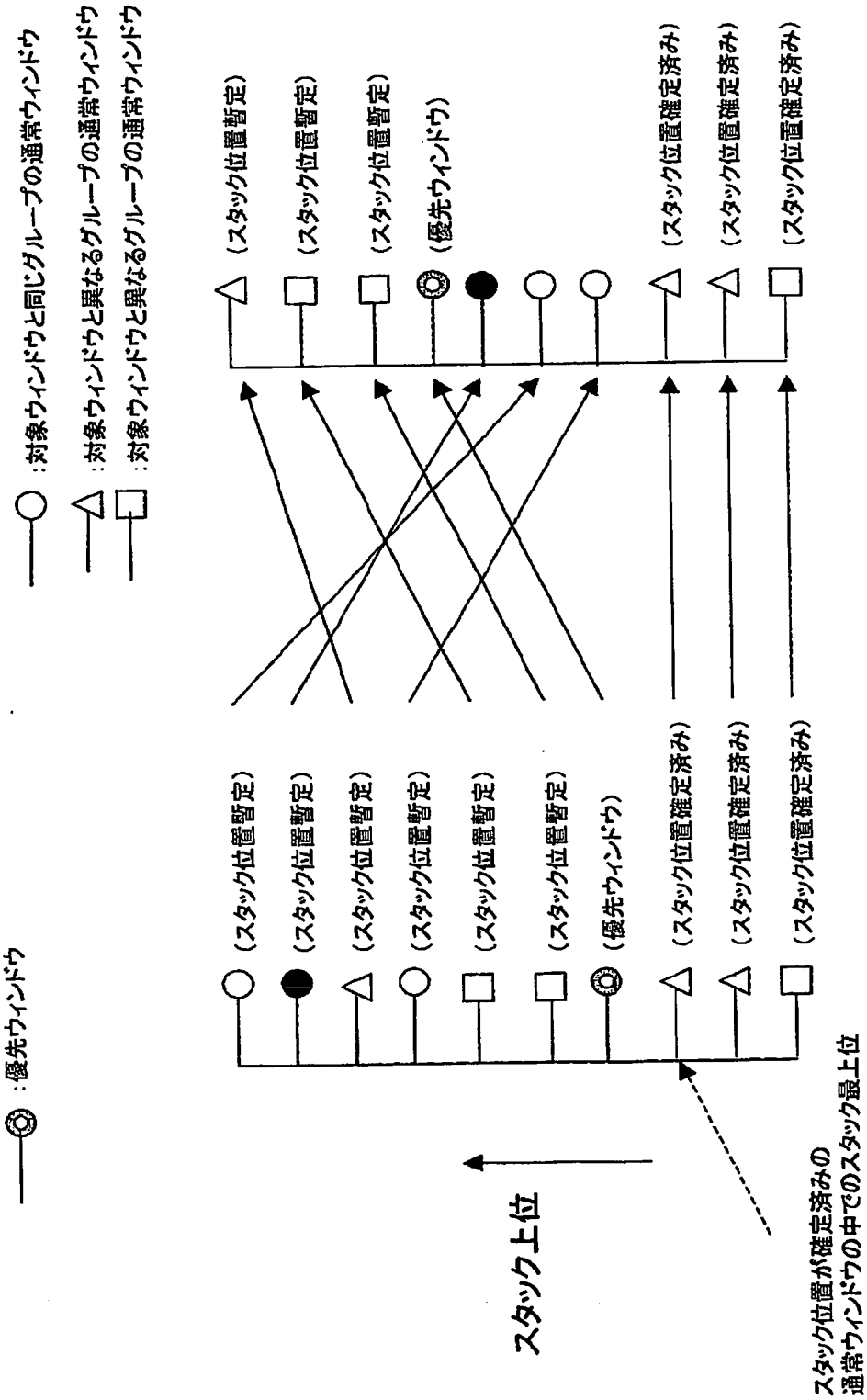
【図７】


:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
:対象ウィンドウである優先ウィンドウ
スタック上位
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更前
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更後

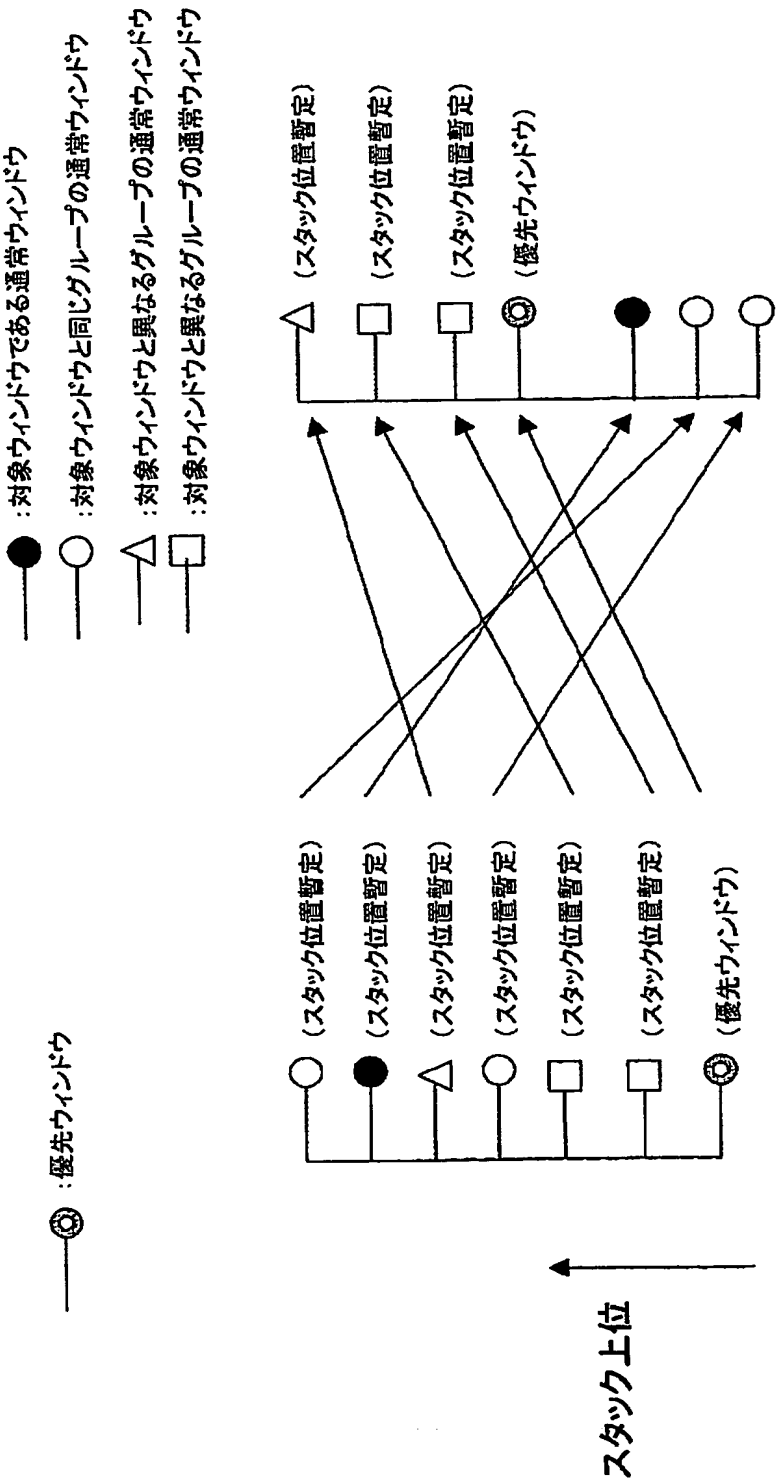
【図８】


:対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
スタック上位
スタック変更前
スタック変更後
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（優先ウィンドウ）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（優先ウィンドウ）
（スタック位置暫定）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）

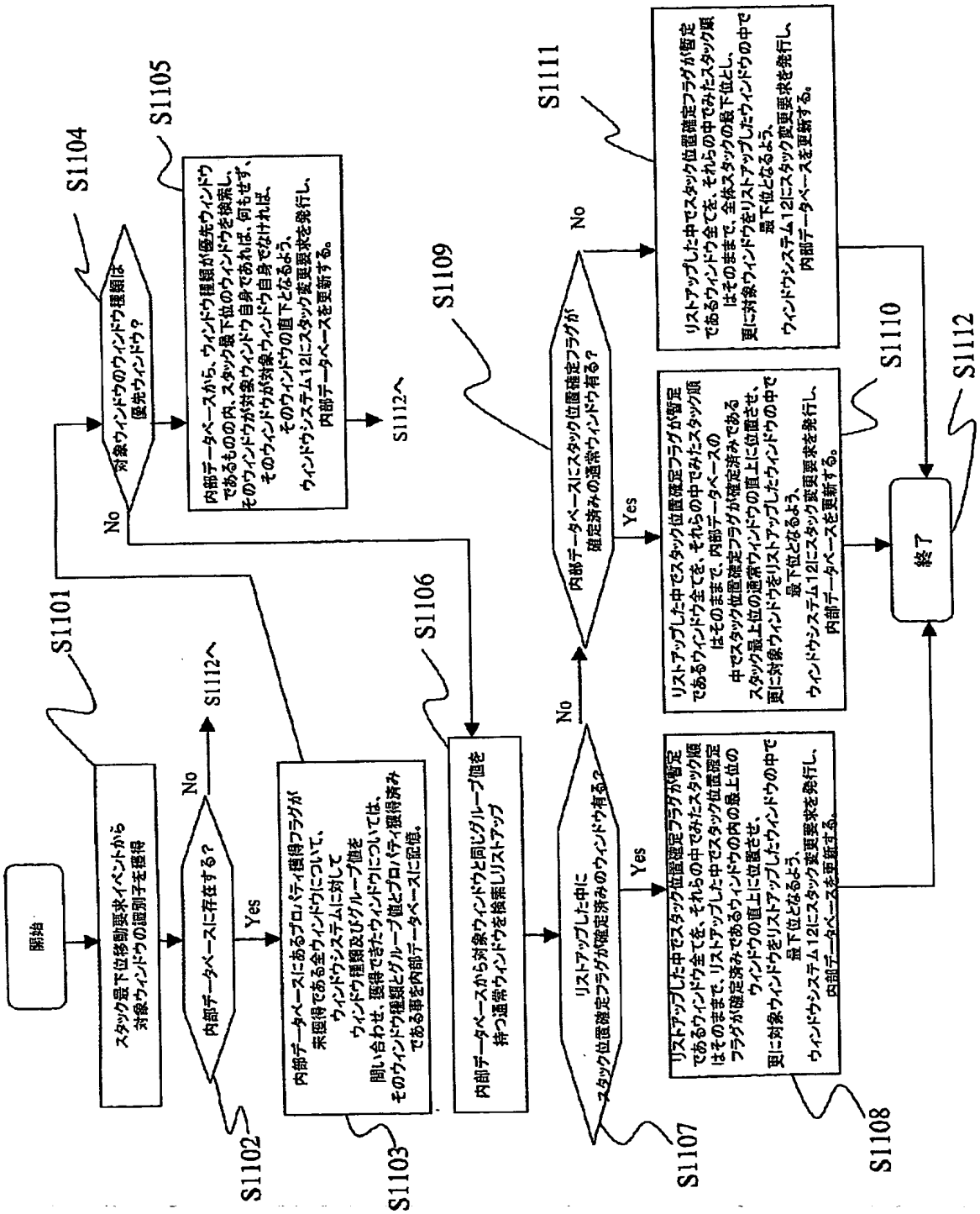
【図９】


:対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
スタック上位
スタック位置が確定済みの
通常ウィンドウの中でのスタック最上位
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更前
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更後

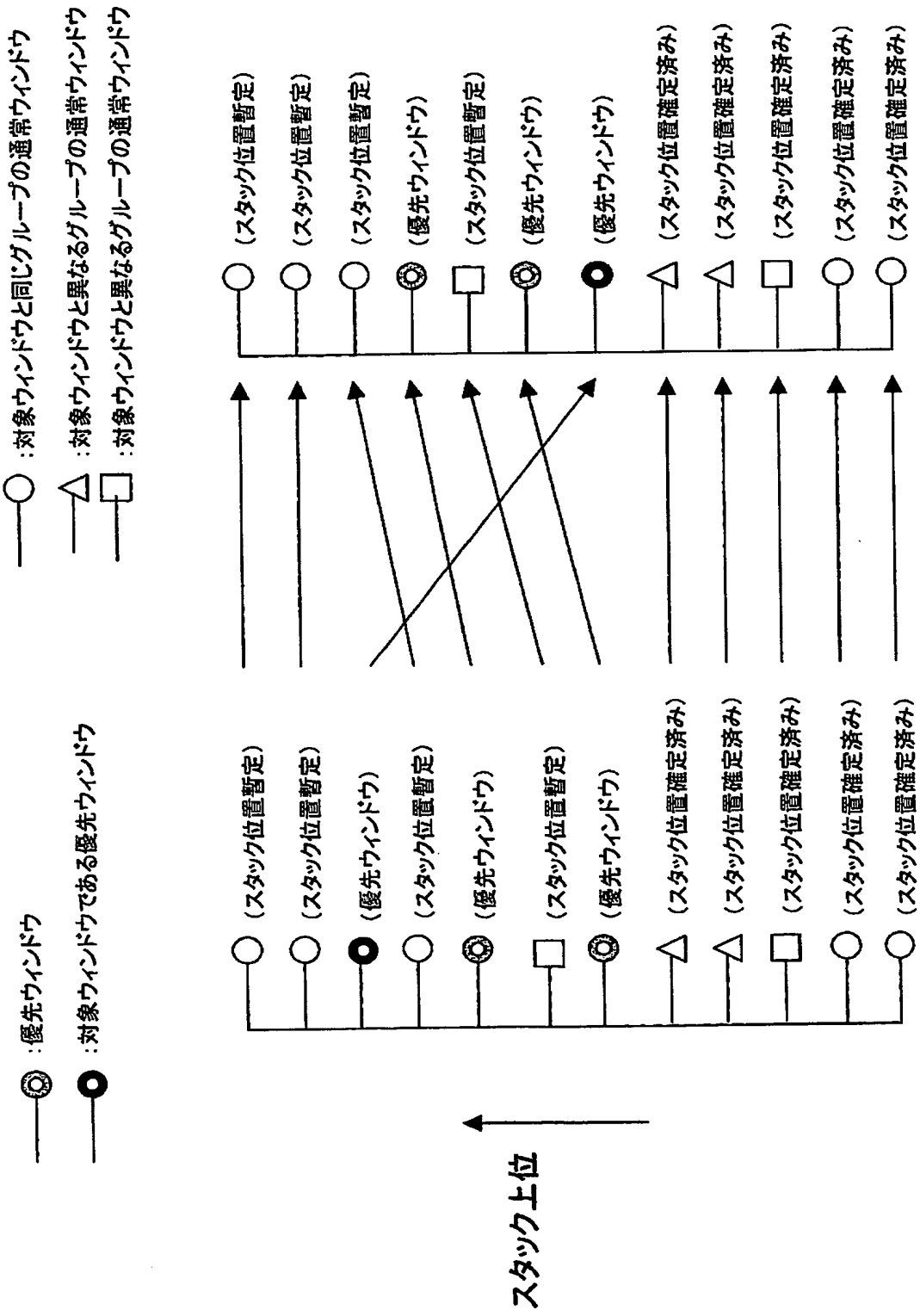
【図１０】


：対象ウィンドウである通常ウィンドウ
：対象ウィンドウと同じグループの通常ウィンドウ
：対象ウィンドウと異なるグループの通常ウィンドウ
：対象ウィンドウと異なるグループの通常ウィンドウ
：優先ウィンドウ
スタック上位
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（優先ウィンドウ）
スタック変更前
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（優先ウィンドウ）
スタック変更後

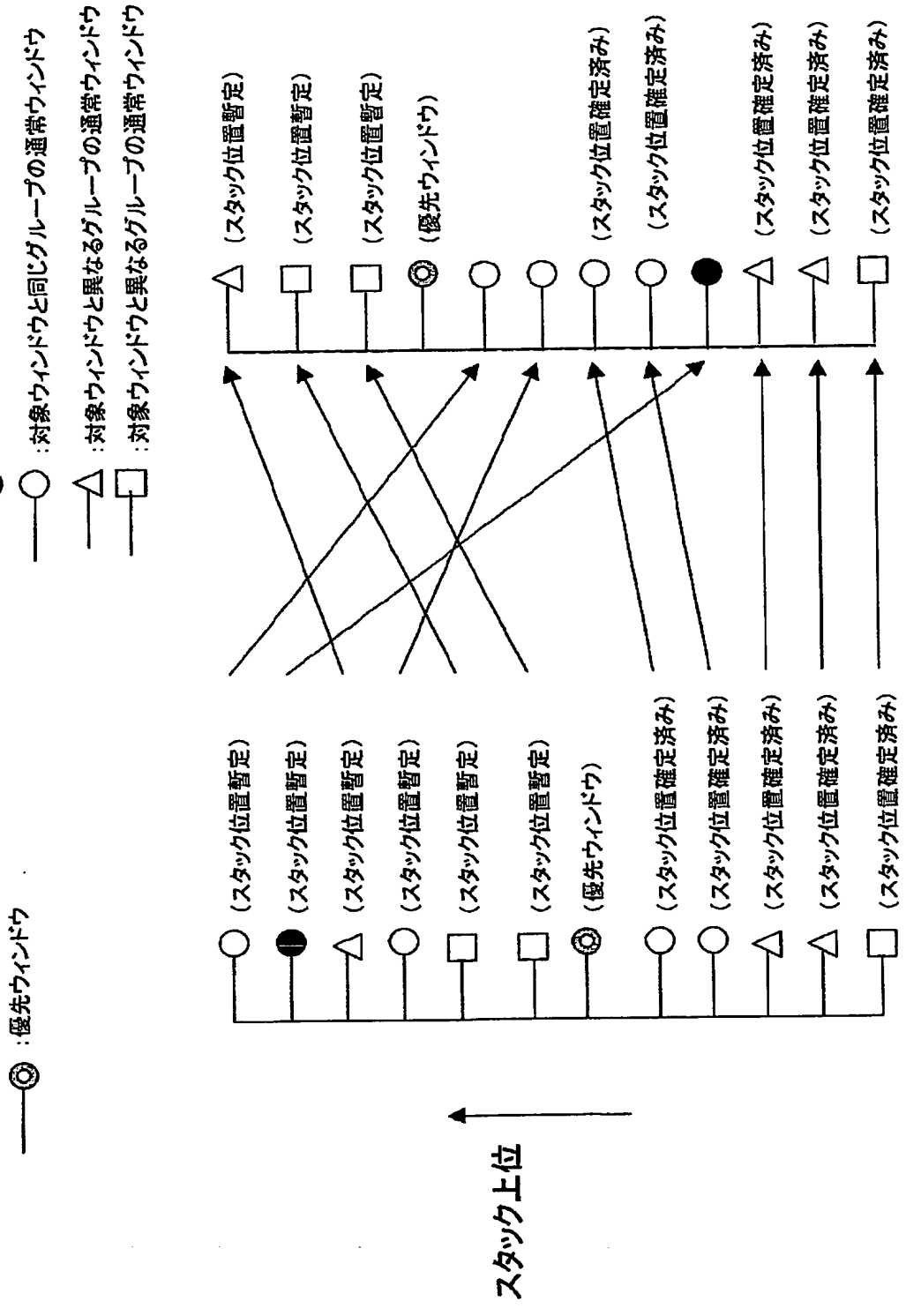
【図１１】


開始
スタック最下位移動要求イベントから
対象ウィンドウの識別子を獲得
S1101
内部データベースに存在する？
Yes
No
S1112へ
S1102
内部データベースにあるプロパティ獲得フラグが
未獲得である全ウィンドウについて、
ウィンドウシステムに対して
ウィンドウ種類及びグループ値を
問い合わせ、獲得できたウィンドウについては、
そのウィンドウ種類とグループ値とプロパティ獲得済み
である事を内部データベースに記憶。
S1103
S1104
対象ウィンドウのウィンドウ種類は
優先ウィンドウ？
No
Yes
S1105
内部データベースから、ウィンドウ種類が優先ウィンドウ
であるものの内、スタック最下位のウィンドウを検索し、
そのウィンドウが対象ウィンドウ自身であれば、何もせず、
そのウィンドウが対象ウィンドウ自身でなければ、
そのウィンドウの直下となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
S1112へ
S1106
内部データベースから対象ウィンドウと同じグループ値を
持つ通常ウィンドウを検索しリストアップ
S1109
内部データベースにスタック位置確定フラグが
確定済みの通常ウィンドウ有る？
No
Yes
S1111
リストアップした中でスタック位置確定フラグが暫定
であるウィンドウ全てを、それらの中でみたスタック順
はそのままで、内部データベースの
中でスタック最上位の通常ウィンドウの直上に位置させ、
更に対象ウィンドウをリストアップしたウィンドウの中で
最下位となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
S1110
リストアップした中でスタック位置確定フラグが暫定
であるウィンドウ全てを、それらの中でみたスタック順
はそのままで、全体スタックの最下位とし、
更に対象ウィンドウをリストアップしたウィンドウの中で
最下位となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
S1112
リストアップした中にスタック位置確定フラグが確定済みのウィンドウ有る？
No
Yes
S1107
リストアップした中でスタック位置確定フラグが暫定
であるウィンドウ全てを、それらの中でみたスタック順
はそのままで、リストアップした中でスタック位置確定
フラグが確定済みであるウィンドウの内の最上位の
ウィンドウの直上に位置させ、
更に対象ウィンドウをリストアップしたウィンドウの中で
最下位となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
S1108
終了

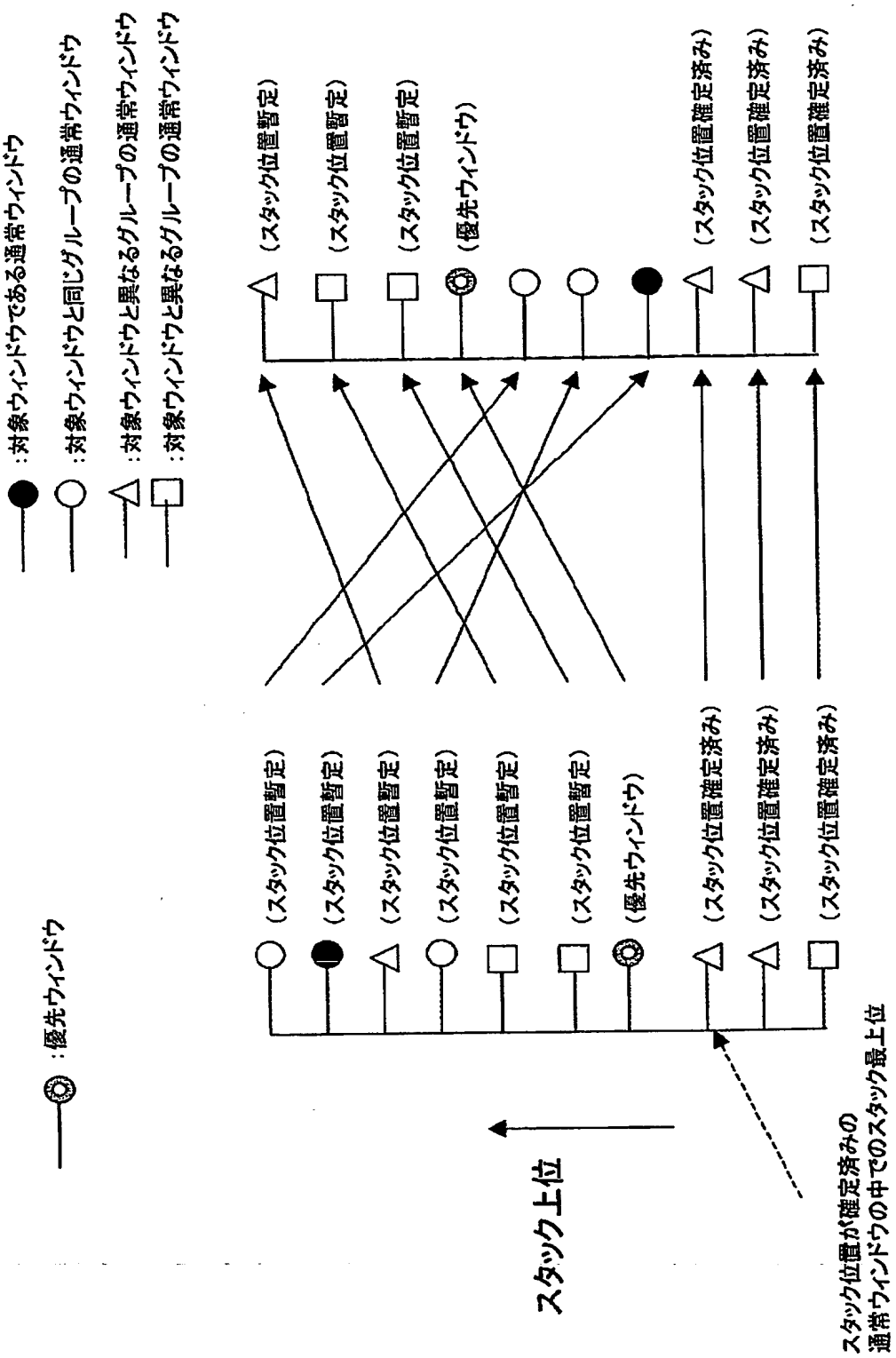
【図１２】


:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
:対象ウィンドウである優先ウィンドウ
スタック上位
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更前
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置暫定)
(優先ウィンドウ)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更後

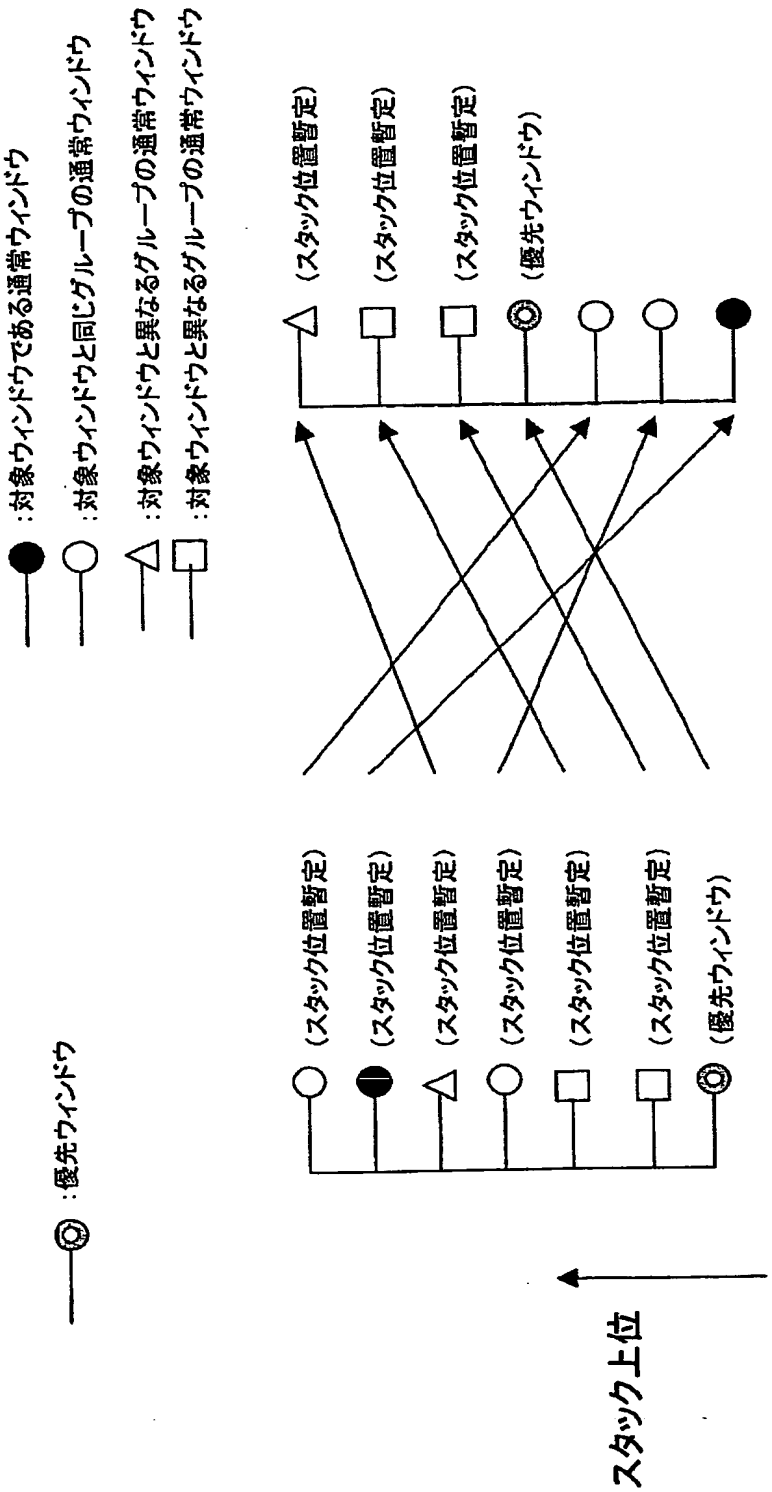
【図１３】


:対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
スタック上位
スタック変更前
スタック変更後
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)

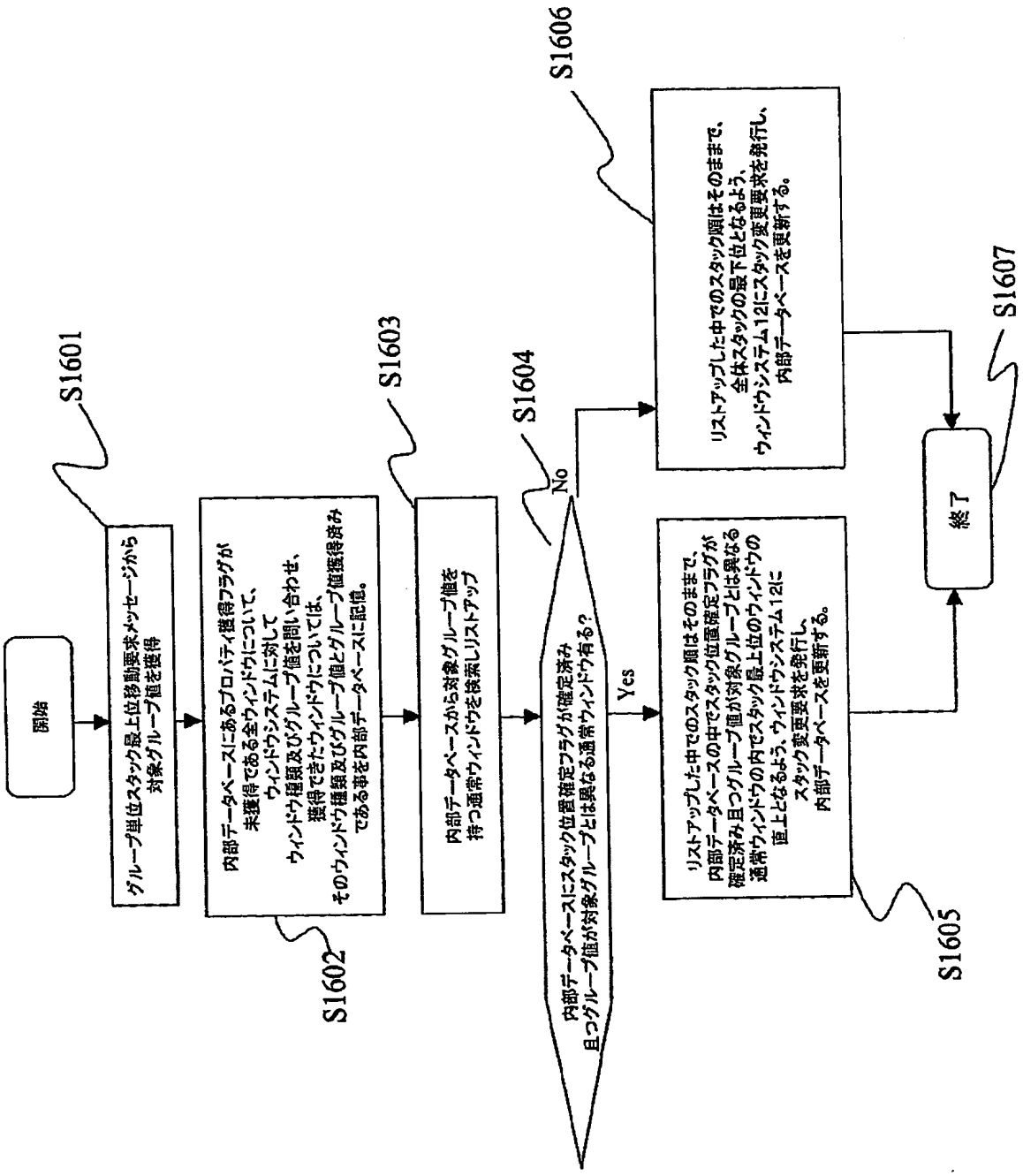
【図１４】


:対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック上位
スタック位置が確定済みの
通常ウィンドウの中でのスタック最上位
スタック変更前
スタック変更後

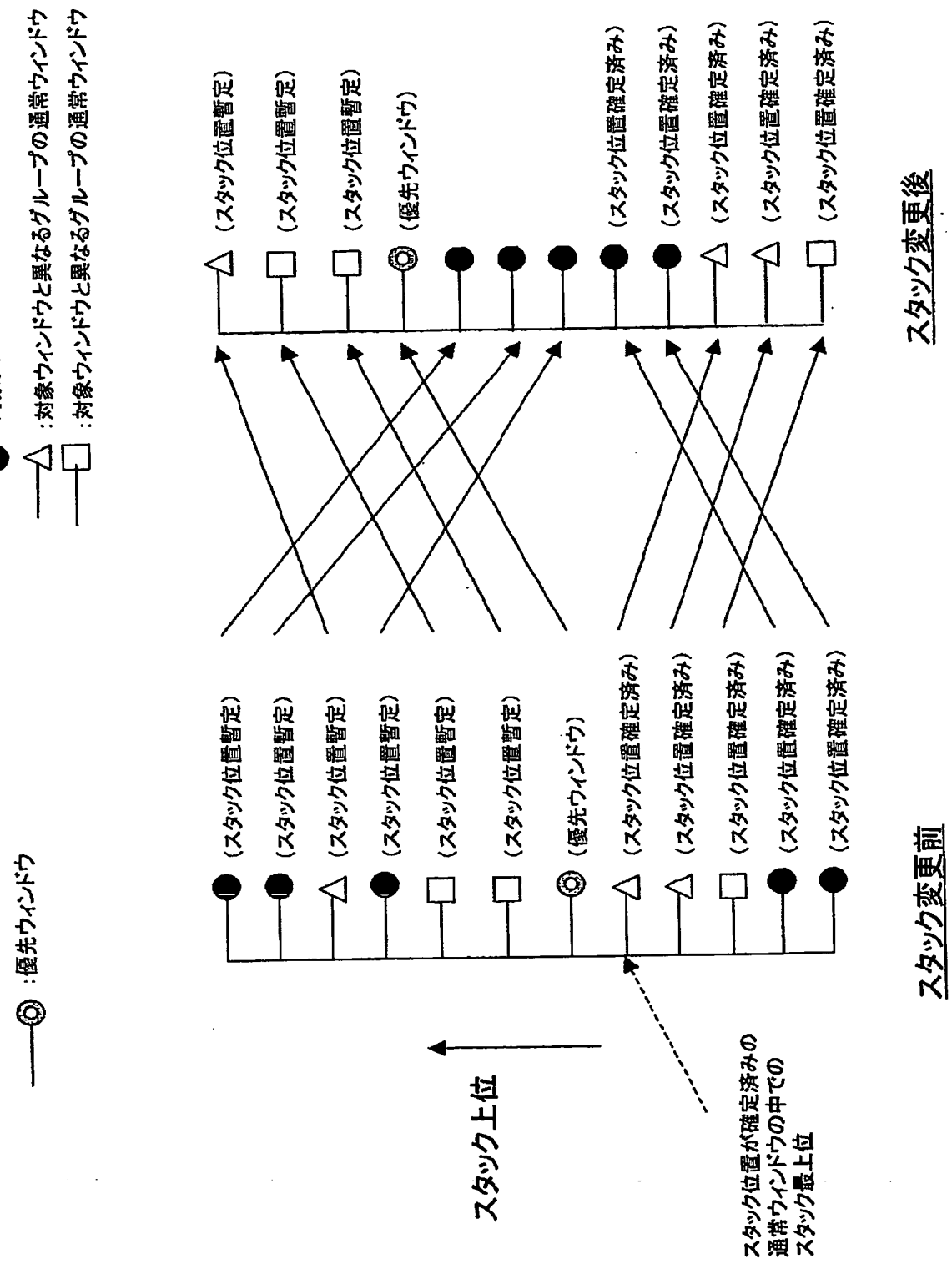
【図１５】


:対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
スタック上位
スタック変更前
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
スタック変更後

【図１６】


開始
グループ単位スタック最上位移動要求メッセージから
対象グループ値を獲得
S1601
内部データベースにあるプロパティ獲得フラグが
未獲得である全ウィンドウについて、
ウィンドウシステムに対して
ウィンドウ種類及びグループ値を問い合わせ、
獲得できたウィンドウについては、
そのウィンドウ種類及びグループ値とグループ値獲得済み
である事を内部データベースに記憶。
S1602
S1603
内部データベースから対象グループ値を
持つ通常ウィンドウを検索しリストアップ
S1604
内部データベースにスタック位置確定フラグが確定済み
且つグループ値が対象グループとは異なる通常ウィンドウ有る?
No
Yes
S1606
リストアップした中でのスタック順はそのままで、
全体スタックの最下位となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
リストアップした中でのスタック順はそのままで、
内部データベースの中でスタック位置確定フラグが
確定済み且つグループ値が対象グループとは異なる
通常ウィンドウの内でスタック最上位のウィンドウの
直上となるよう、ウィンドウシステム12に
スタック変更要求を発行し、
内部データベースを更新する。
S1605
終了
S1607

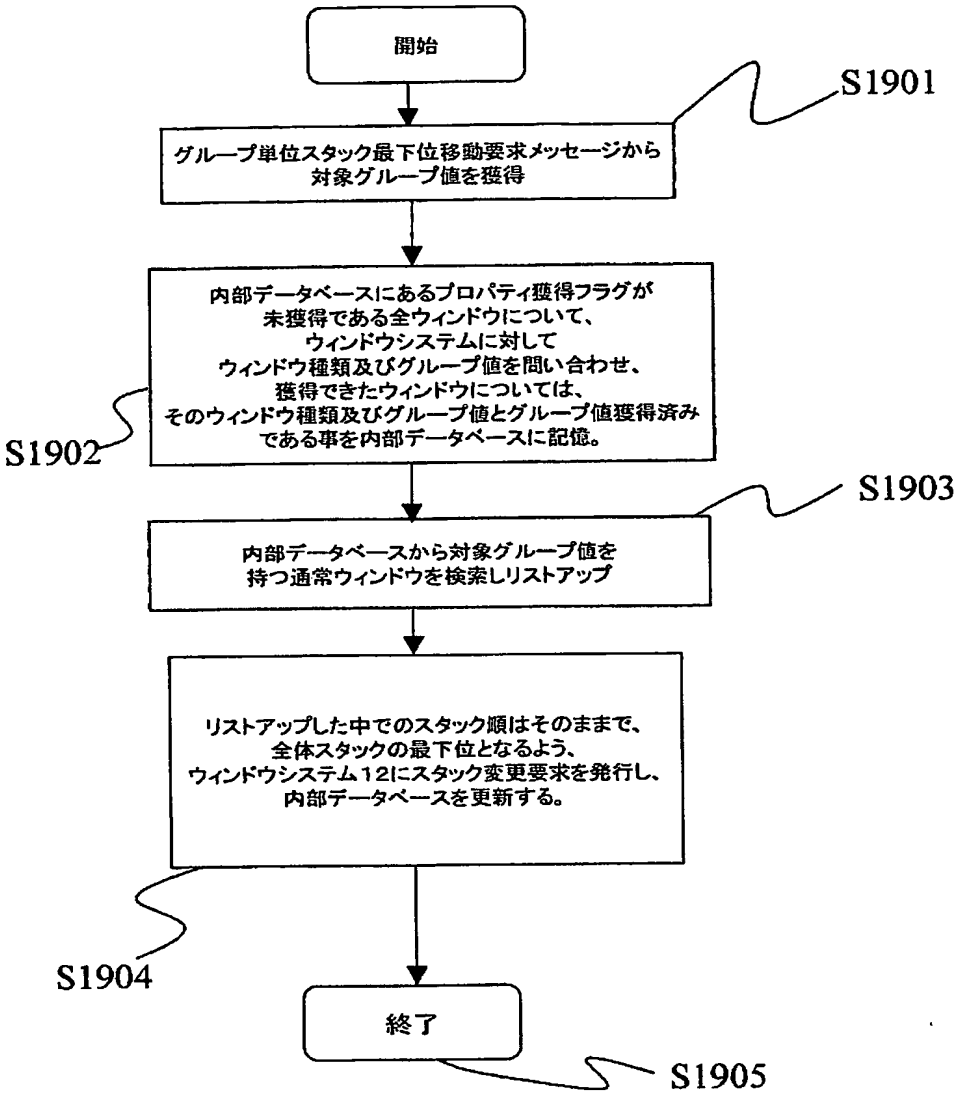
【図１７】


：対象グループ値を持つ通常ウィンドウ
：対象ウィンドウと異なるグループの通常ウィンドウ
：対象ウィンドウと異なるグループの通常ウィンドウ
：優先ウィンドウ
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（優先ウィンドウ）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
スタック上位
スタック位置が確定済みの
通常ウィンドウの中での
スタック最上位
スタック変更前
スタック変更後

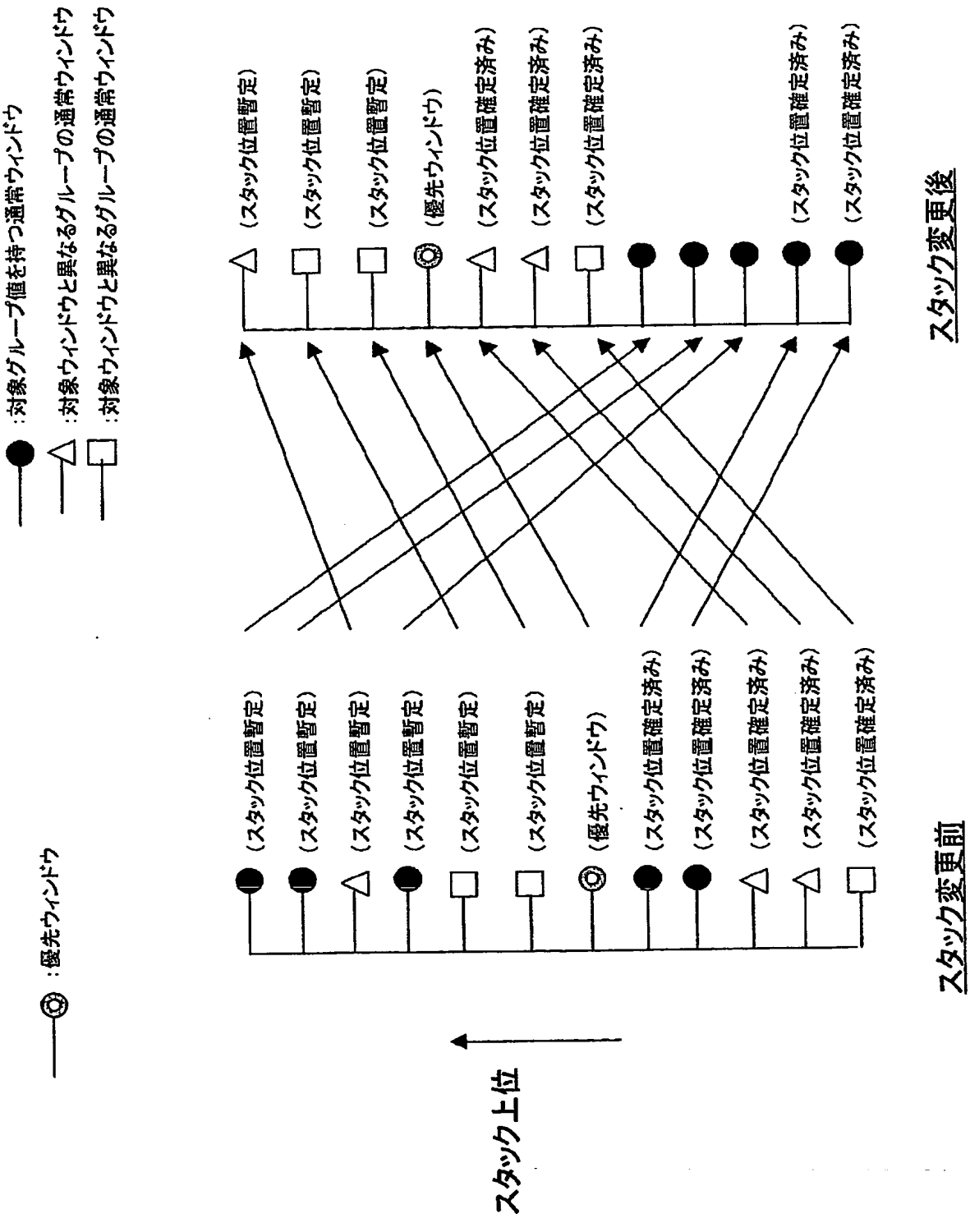
【図１８】

:対象グループ値を持つ通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
スタック上位
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
スタック変更前
スタック変更後

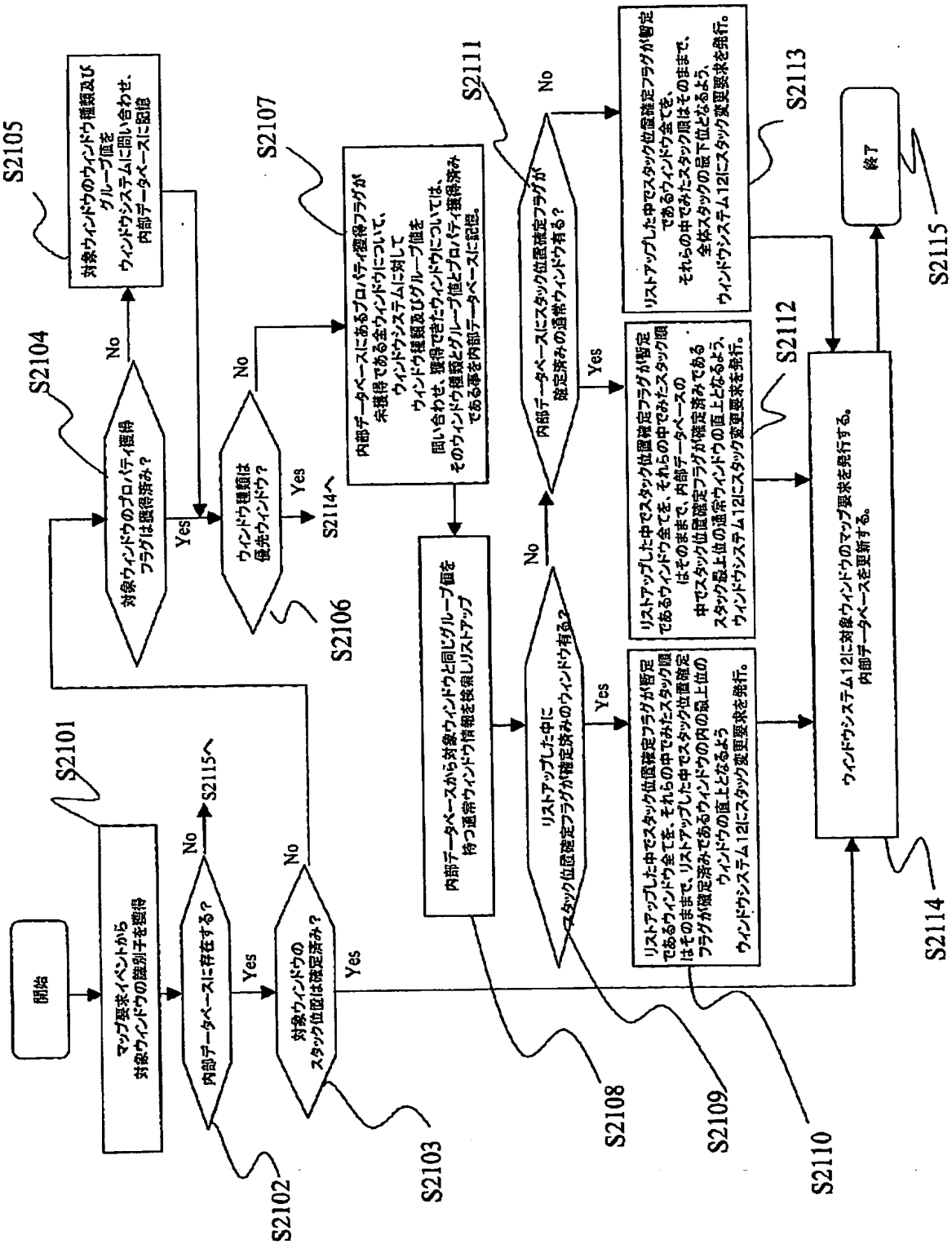
【図１９】


開始
S1901
グループ単位スタック最下位移動要求メッセージから
対象グループ値を獲得
内部データベースにあるプロパティ獲得フラグが
未獲得である全ウィンドウについて、
ウィンドウシステムに対して
ウィンドウ種類及びグループ値を問い合わせ、
獲得できたウィンドウについては、
そのウィンドウ種類及びグループ値とグループ値獲得済み
である事を内部データベースに記憶。
S1902
S1903
内部データベースから対象グループ値を
持つ通常ウィンドウを検索しリストアップ
リストアップした中でのスタック順はそのままで、
全体スタックの最下位となるよう、
ウィンドウシステム12にスタック変更要求を発行し、
内部データベースを更新する。
S1904
終了
S1905

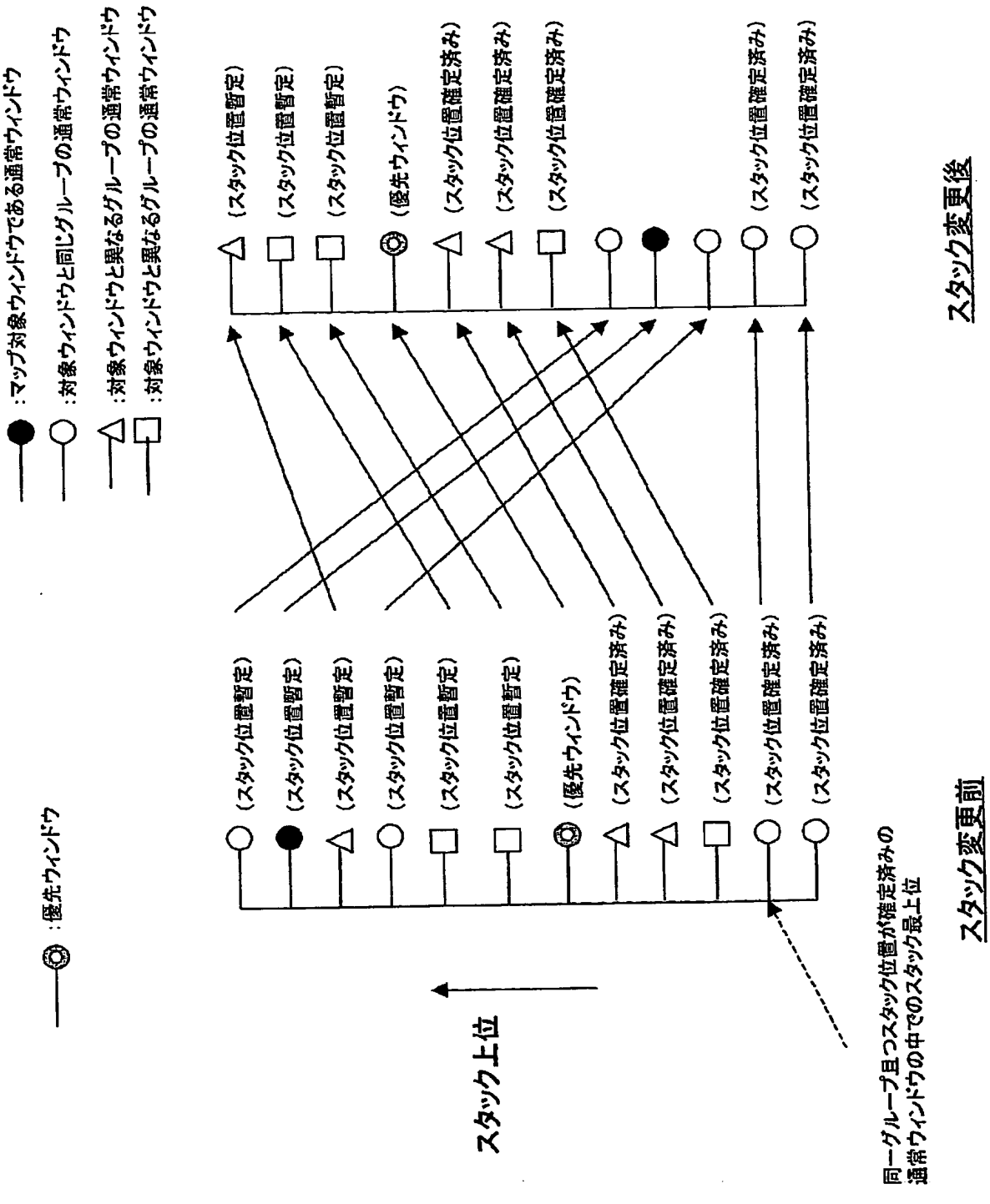
【図２０】


:対象グループ値を持つ通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
スタック上位
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更前
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更後

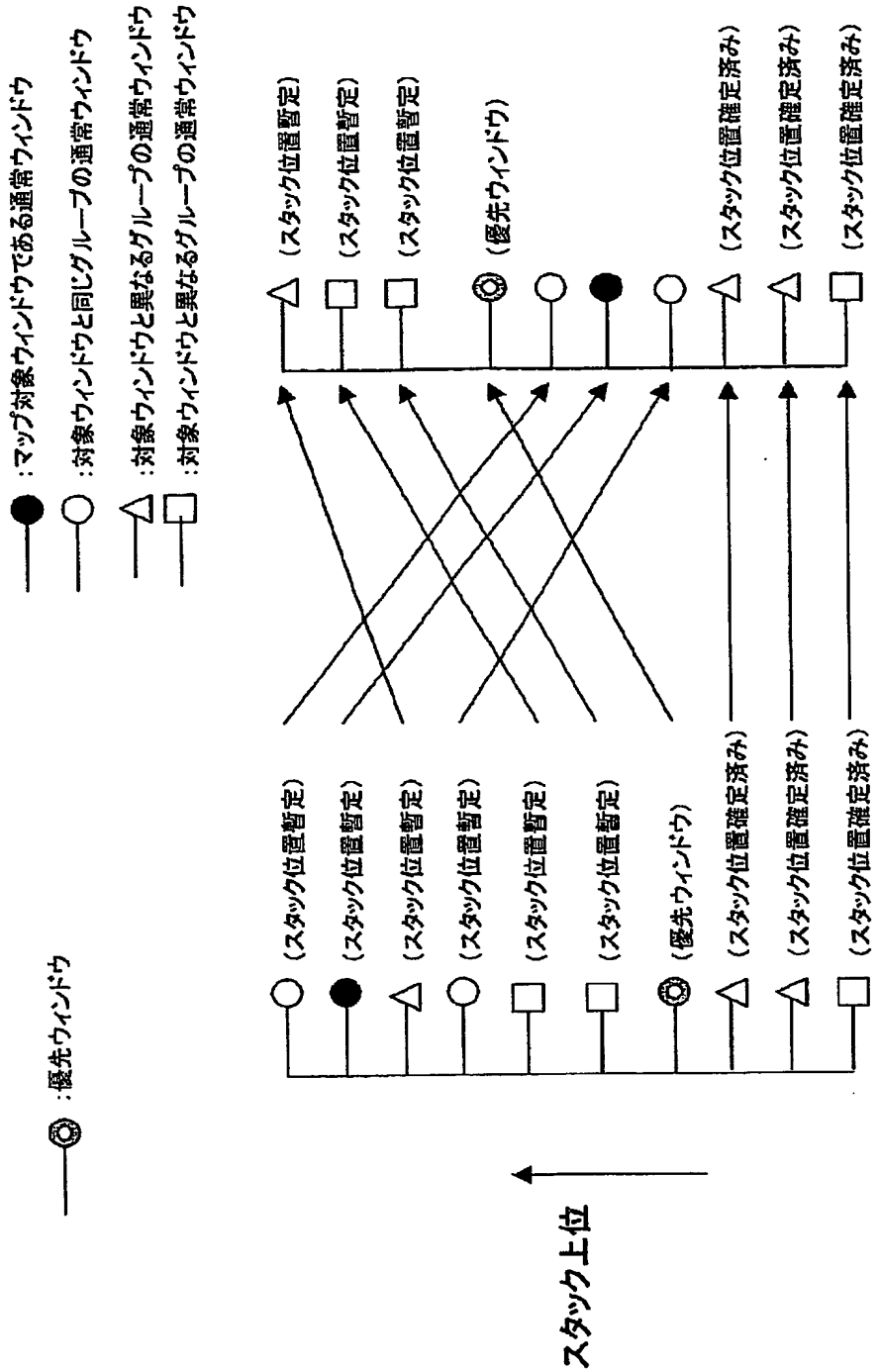
【図２１】


開始
マップ要求イベントから
対象ウィンドウの識別子を獲得
S2101
内部データベースに存在する？
No
Yes
S2115へ
S2102
対象ウィンドウの
スタック位置は確定済み？
No
Yes
S2103
対象ウィンドウのプロパティ獲得
フラグは獲得済み？
No
Yes
S2104
対象ウィンドウのウィンドウ種類及び
グループ値を
ウィンドウシステムに問い合わせ、
内部データベースに記憶
S2105
ウィンドウ種類は
優先ウィンドウ？
No
Yes
S2114へ
S2106
内部データベースにあるプロパティ獲得フラグが
未獲得である全ウィンドウについて、
ウィンドウシステムに対して
ウィンドウ種類及びグループ値を
問い合わせ、獲得できたウィンドウについては、
そのウィンドウ種類とグループ値とプロパティ獲得済み
である事を内部データベースに記憶。
S2107
内部データベースから対象ウィンドウと同じグループ値を
持つ通常ウィンドウ情報を検索しリストアップ
S2108
リストアップした中に
スタック位置確定フラグが確定済みのウィンドウ有る？
No
Yes
S2109
リストアップした中でスタック位置確定フラグが暫定
であるウィンドウ全てを、それらの中でみたスタック順
はそのままで、リストアップした中でスタック位置確定
フラグが確定済みであるウィンドウの内の最上位の
ウィンドウの直上となるよう
ウィンドウシステム12にスタック変更要求を発行。
S2110
内部データベースにスタック位置確定フラグが
確定済みの通常ウィンドウ有る？
No
Yes
S2111
リストアップした中でスタック位置確定フラグが暫定
であるウィンドウ全てを、それらの中でみたスタック順
はそのままで、内部データベースの
中でスタック位置確定フラグが確定済みである
スタック最上位の通常ウィンドウの直上となるよう、
ウィンドウシステム12にスタック変更要求を発行。
S2112
リストアップした中でスタック位置確定フラグが暫定
であるウィンドウ全てを、
それらの中でみたスタック順は全てのままで、
全体スタックの最下位となるよう、
ウィンドウシステム12にスタック変更要求を発行。
S2113
ウィンドウシステム12に対象ウィンドウのマップ要求を発行する。
内部データベースを更新する。
S2114
終了
S2115

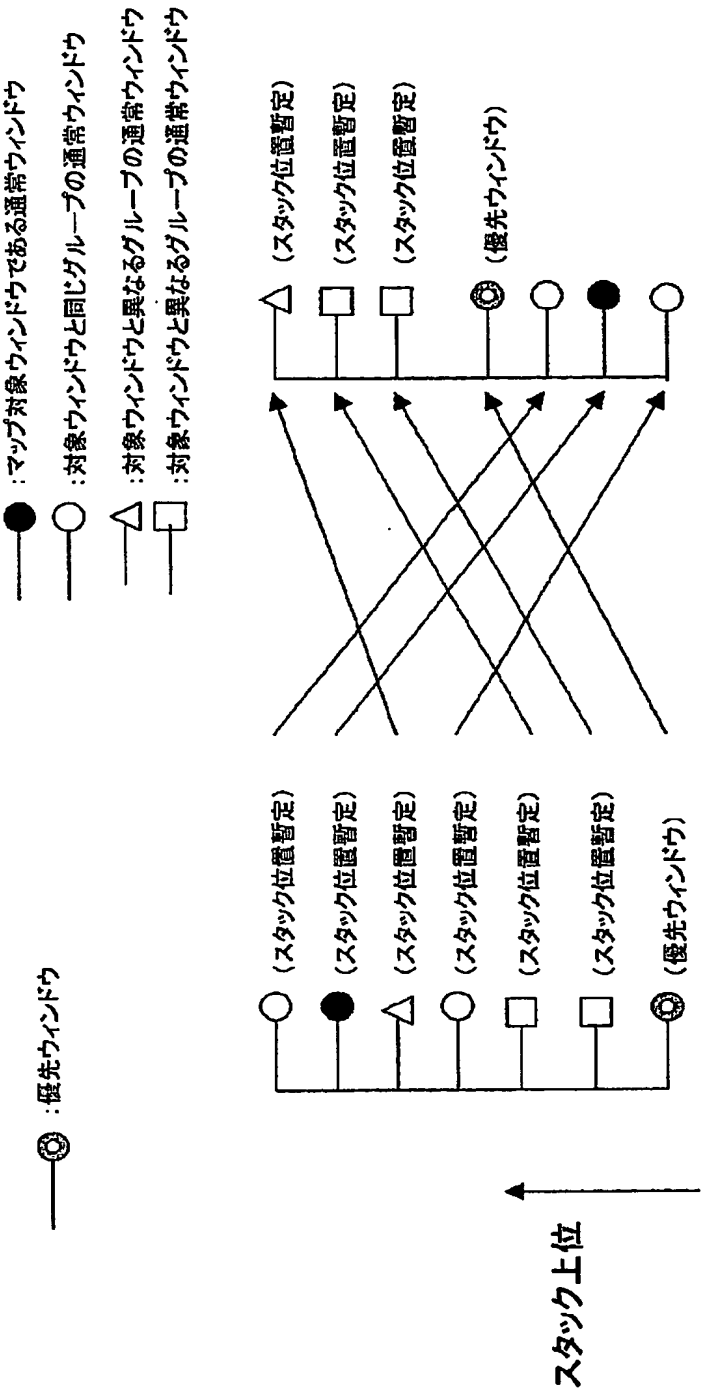
【図２２】


:マップ対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
スタック上位
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更前
スタック変更後
同一グループ且つスタック位置が確定済みの
通常ウィンドウの中でのスタック最上位

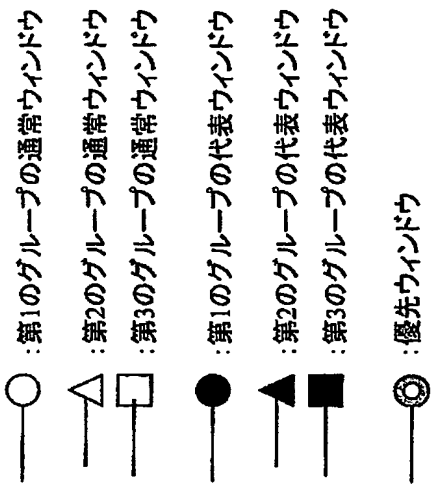
【図２３】

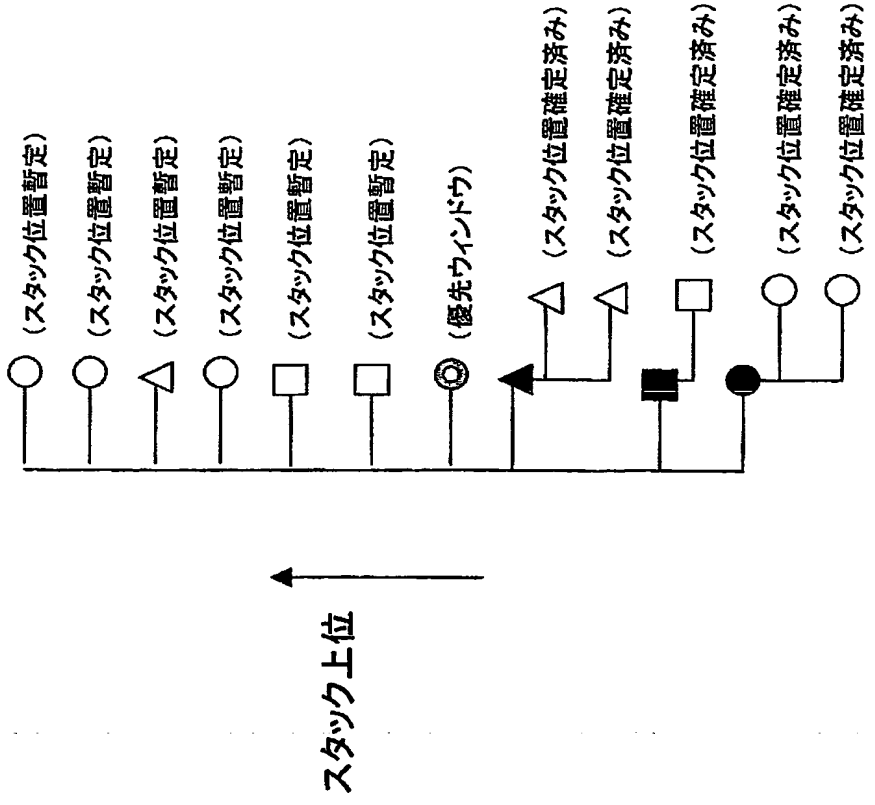

:マップ対象ウィンドウである通常ウィンドウ
:対象ウィンドウと同じグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:対象ウィンドウと異なるグループの通常ウィンドウ
:優先ウィンドウ
スタック上位
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(スタック位置暫定)
(優先ウィンドウ)
(スタック位置確定済み)
(スタック位置確定済み)
(スタック位置確定済み)
スタック変更前
スタック変更後

【図２４】

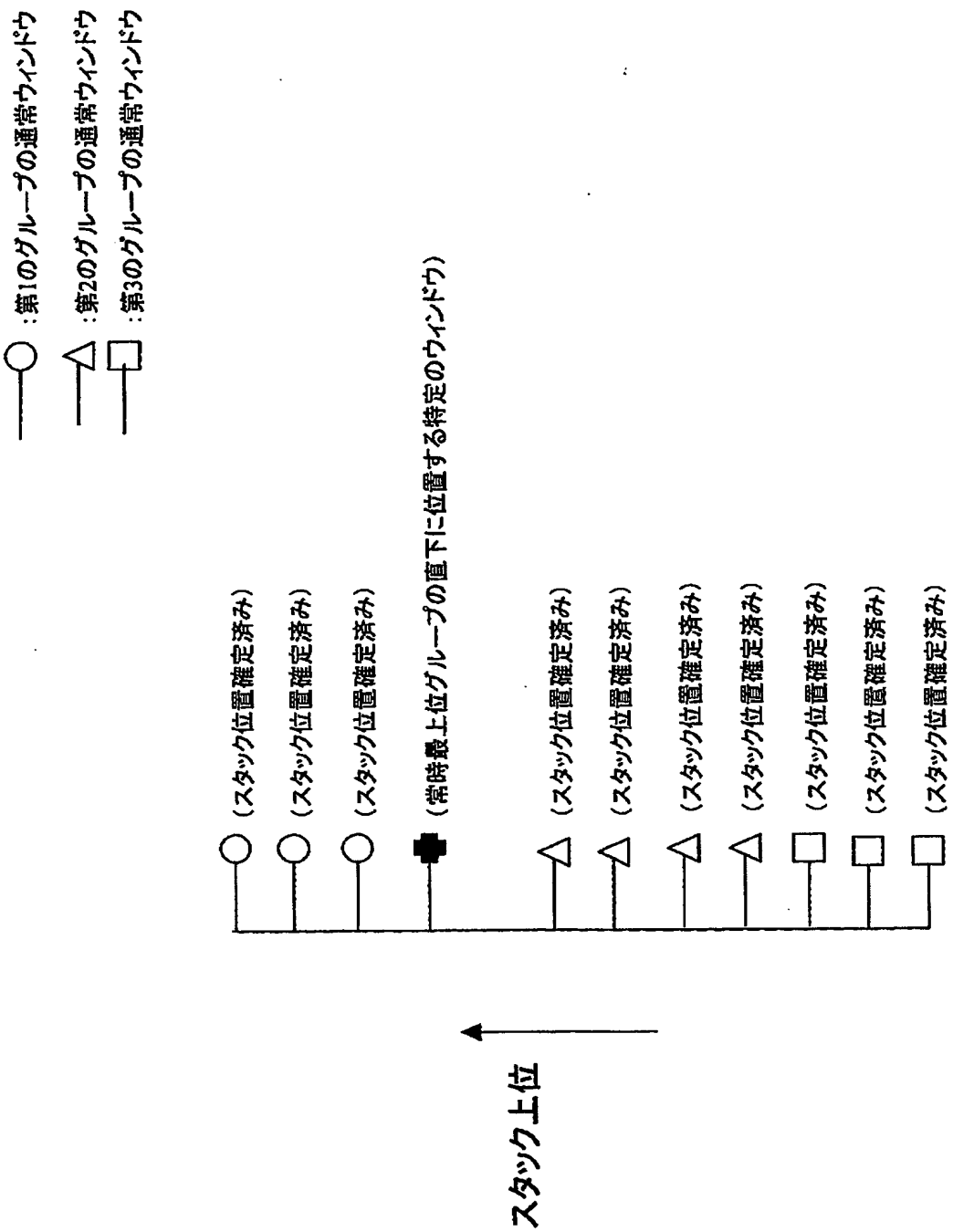

：マップ対象ウィンドウである通常ウィンドウ
：対象ウィンドウと同じグループの通常ウィンドウ
：対象ウィンドウと異なるグループの通常ウィンドウ
：対象ウィンドウと異なるグループの通常ウィンドウ
：優先ウィンドウ
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（優先ウィンドウ）
スタック上位
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（優先ウィンドウ）
スタック変更前
スタック変更後

【図２５】

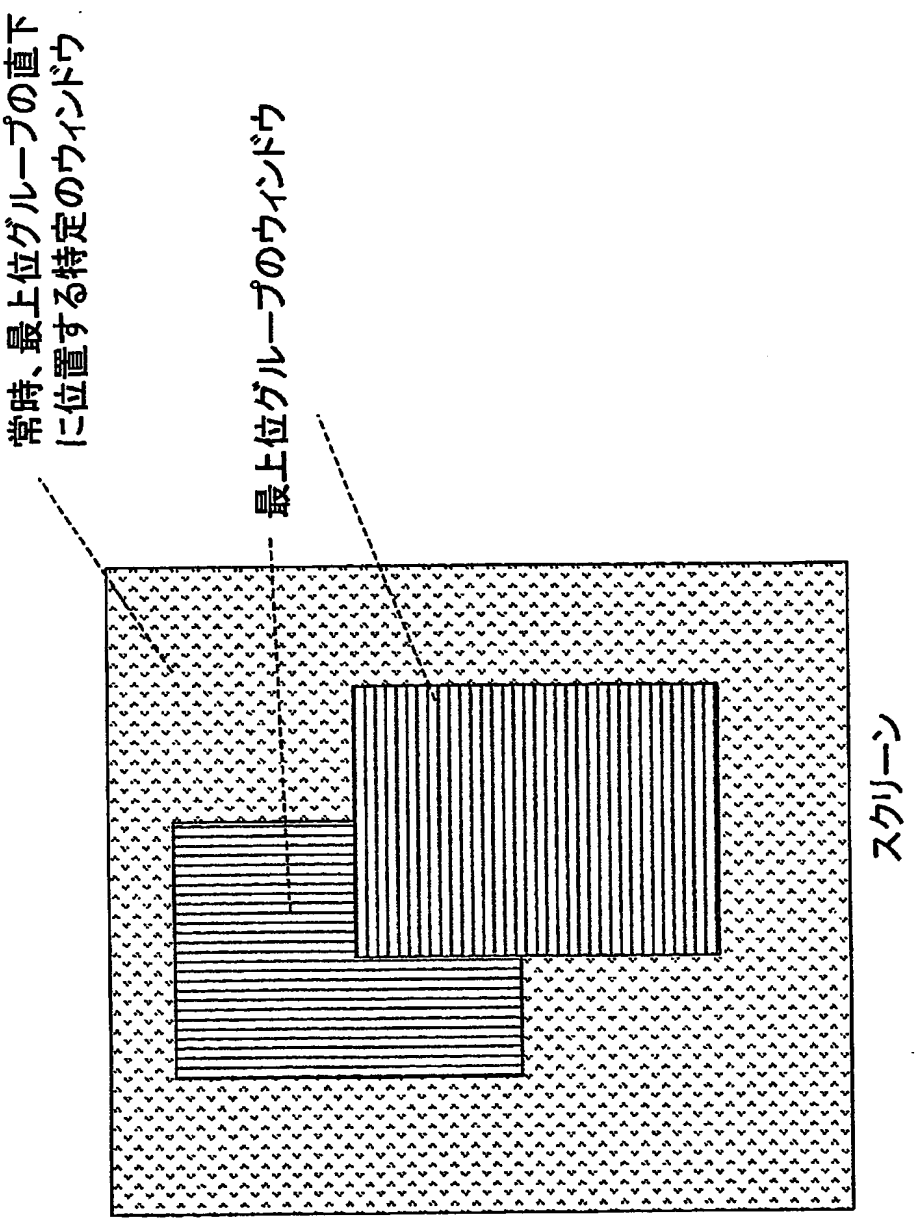

:第1のグループの通常ウィンドウ
:第2のグループの通常ウィンドウ
:第3のグループの通常ウィンドウ
:第1のグループの代表ウィンドウ
:第2のグループの代表ウィンドウ
:第3のグループの代表ウィンドウ
:優先ウィンドウ


（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（スタック位置暫定）
（優先ウィンドウ）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
スタック上位

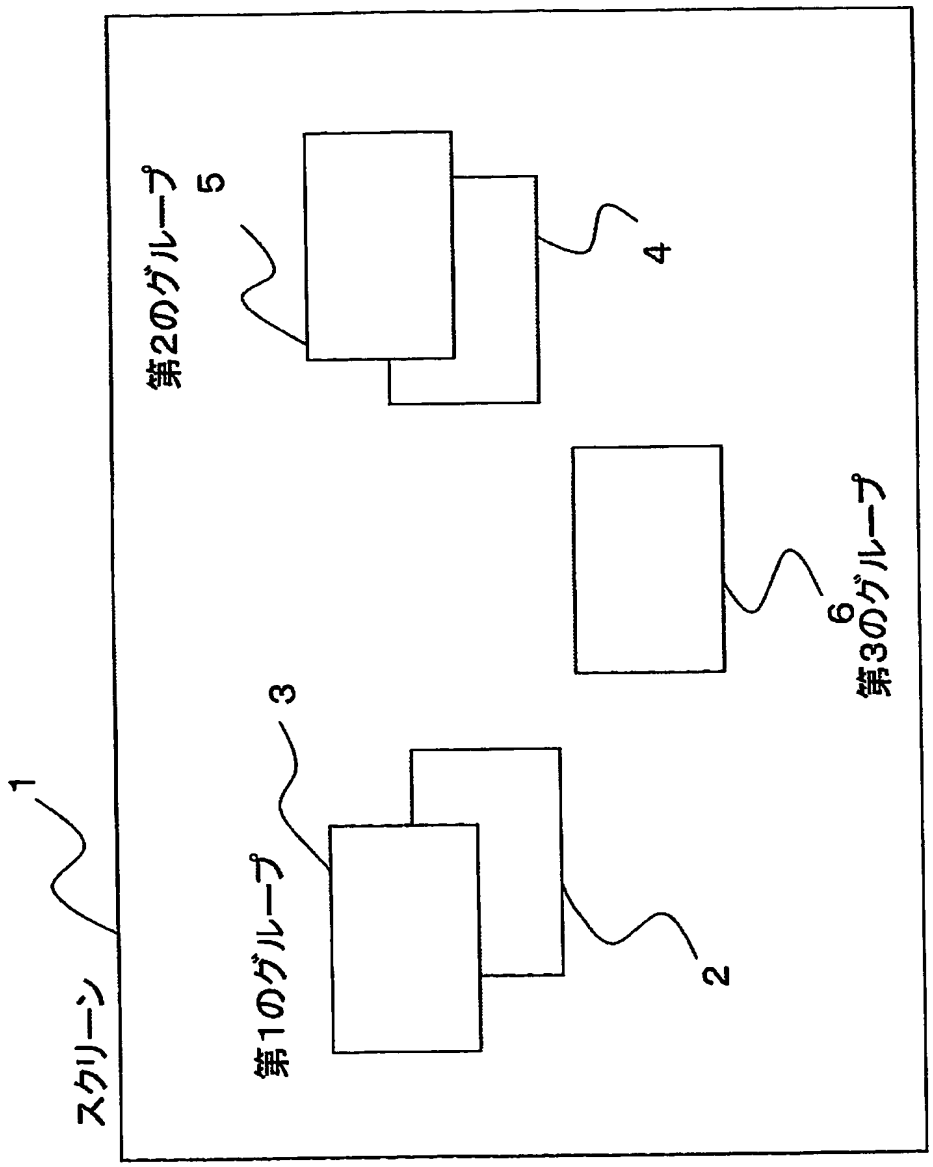
【図２６】


○：第1のグループの通常ウィンドウ
△：第2のグループの通常ウィンドウ
□：第3のグループの通常ウィンドウ
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（常時最上位グループの直下に位置する特定のウィンドウ）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
（スタック位置確定済み）
スタック上位

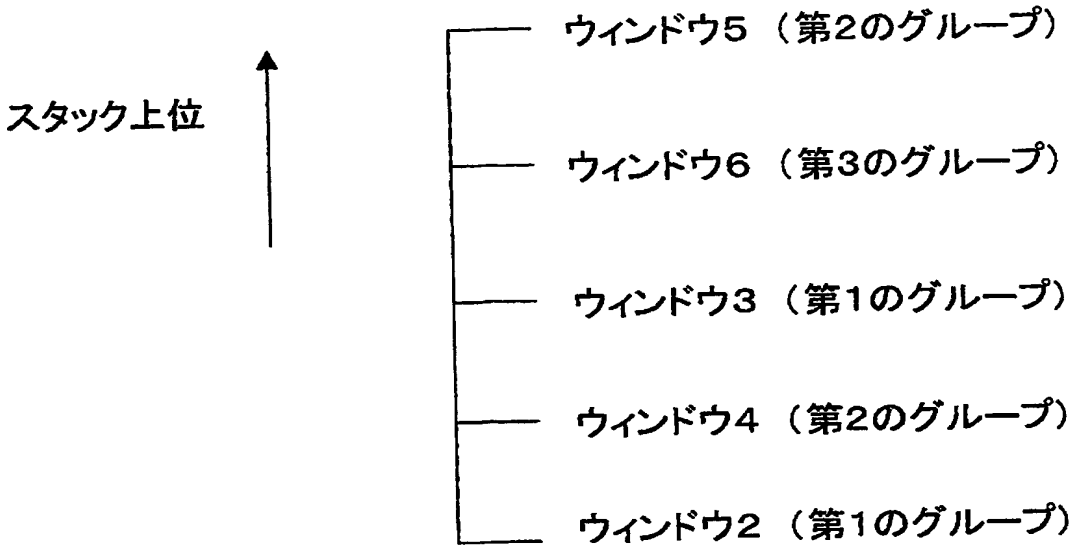
【図２７】



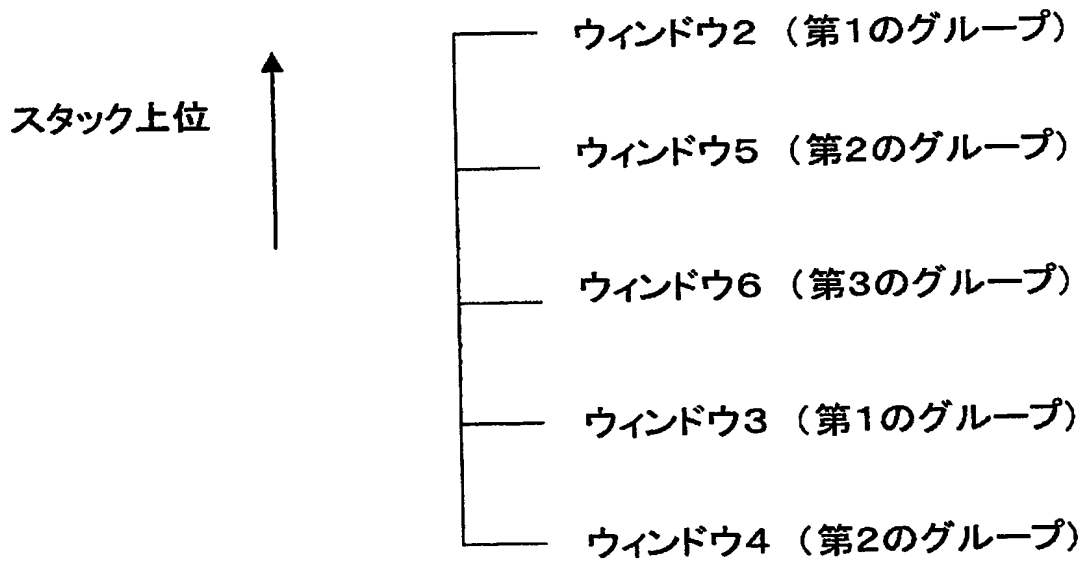
【図２８】



【図２９】


スタック上位
ウィンドウ５（第2のグループ）
ウィンドウ６（第3のグループ）
ウィンドウ３（第1のグループ）
ウィンドウ４（第2のグループ）
ウィンドウ２（第1のグループ）

【図３０】


スタック上位
ウィンドウ２（第1のグループ）
ウィンドウ５（第2のグループ）
ウィンドウ６（第3のグループ）
ウィンドウ３（第1のグループ）
ウィンドウ４（第2のグループ）

【書類名】　要約書

【要約】

【課題】　ウィンドウスタックの制御において、最前面でウィンドウを表示中のアプリケーションのウィンドウ表示に、他のアプリケーションが不用意に影響を及ぼす事の無いウィンドウスタック制御方法を提供することを目的とする。

【解決手段】　コンピュータから表示装置に複数のウィンドウを表示する際のウィンドウの重ね合わせを管理するウィンドウスタックを制御するに際し、アプリケーションプログラムがウィンドウにグループを指定し、ウィンドウ管理プログラムがウィンドウのスタック順をグループ毎に一連とする。

【選択図】　図１

特願２００３－１０６３９３

出　願　人　履　歴　情　報

識別番号　　[０００００５８２１]

１．変更年月日　　１９９０年　８月２８日
[変更理由]　　新規登録
住　所　　大阪府門真市大字門真１００６番地
氏　名　　松下電器産業株式会社